

THESIS / THÈSE

MASTER EN SCIENCES MATHÉMATIQUES

Séries temporelles et réseaux

Botterman, Hong-Lan

Award date:
2017

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



UNIVERSITE DE NAMUR

Faculté des Sciences

Séries temporelles et réseaux

Promoteur : Timoteo CARLETTI

**Mémoire présenté pour l'obtention
du grade académique de master en Sciences mathématiques à finalité spécialisée**

Hong-Lan BOTTERMAN

Juin 2017

Séries temporelles et réseaux

Résumé

Étudier l'évolution temporelle d'un système ainsi que les interactions entre ses composants sont deux moyens de caractériser et comprendre son fonctionnement interne. Pour ce faire, il est courant d'analyser des séries temporelles. Dans ce travail, nous développons deux méthodes permettant d'étudier des séries temporelles sous un autre angle que sont les réseaux. Nous expliquons également deux méthodes permettant d'établir les interactions entre les composants d'un système. L'objectif de ce mémoire était de comprendre les quatre méthodes évoquées ci-dessus, de reproduire les résultats présents dans certains articles de référence et d'analyser, si possible, les résultats obtenus.

Mots-clefs : série temporelle, réseau, information.

Abstract

Studying the temporal evolution of a system as well as the interactions between its components are two ways for characterizing and understanding its internal working. To do this, it is common to analyze time series. In this work, we develop two methods allowing to study time series from a different point of view which are networks. We also explain two other methods for establishing the interactions between a system's components. The aim of this master's thesis was therefore to understand the methods previously mentioned, reproduce some results presented in some papers and analyze, if possible, the obtained results.

Keywords : time series, networks, information.

Je remercie mon promoteur Timoteo Carletti pour son accompagnement efficace tout au long de ce travail. Je remercie également Stefanella Boatto pour m'avoir fourni certaines données. Je voudrais également remercier Alexandre Mauroy, Renaud Lambiotte et Jean-Yves Gnabo pour le temps consacré à lire ce travail. De façon plus générale, merci à toutes les personnes qui ont participé, de près ou de loin, à l'élaboration de ce travail.

Table des matières

Introduction	1
1 Rappels de théorie de graphes et représentation	2
1.1 Définitions	2
1.2 Outil de représentation	2
2 Méthode des quantiles	5
2.1 Aspects théoriques	5
2.2 Quelques résultats	8
2.2.1 Séries périodiques et aléatoires	8
2.2.2 Rossler & Lorenz	12
2.2.3 Chaîne de Markov d'ordre supérieur	21
2.3 Conclusion	26
3 Graphes de visibilité	27
3.1 Aspects théoriques	27
3.1.1 Graphe de visibilité naturelle	27
3.1.2 Graphe de visibilité horizontale	27
3.1.3 Graphe de visibilité horizontale dirigé	30
3.2 Quelques résultats	31
3.2.1 Estimation de l'exposant de Hurst avec NVg	31
3.2.2 Filtrage de bruit avec HVg	34
3.2.3 Mesure de l'irréversibilité avec DHVg	37
3.3 Conclusion	42
4 Taux d'information mutuelle	44
4.1 Aspects théoriques	44
4.1.1 Entropie de Shannon	44
4.1.2 Information mutuelle	45
4.1.3 Taux d'information mutuelle	46
4.1.4 Inférence de réseau avec MIR	47
4.2 Quelques résultats	48
4.2.1 Modèles	48
4.2.2 Forces de couplage	49
4.2.3 Nombre de bins	50
4.2.4 Longueur des séries temporelles et bruit	52
4.2.5 MI versus MIR	53
4.2.6 Autre Histogramme	54
4.3 Conclusion	56
5 Dynamical Cluster Index	57
5.1 Aspects théoriques	57
5.2 Quelques résultats	58
5.2.1 Réseaux booléens	58
5.2.2 Dynamique <i>leader-follower</i>	60

5.2.3	Modèle logistique	62
5.3	Conclusion	63
6	Application	65
6.1	Virus de la dengue	65
6.2	Données cerveau	71
	Conclusion	77
	Références - Annexes	78

Introduction

Étudier l'évolution temporelle d'un système ainsi que les interactions entre ses composants sont deux moyens de caractériser et comprendre son fonctionnement interne. Pour ce faire, il est courant d'analyser des séries temporelles. Sans tenir compte d'aucun processus sous-jacent, ces dernières se définissent simplement comme des ensembles ordonnés de valeurs. Plusieurs techniques allant des méthodes de temps-fréquences à des méthodes non linéaires permettent de caractériser une série temporelle. Parmi les méthodes de temps-fréquence, il y a par exemple, la transformée de Fourier [53] et plus particulièrement, la transformée de Fourier à court terme [43] ainsi que les ondelettes [60]. Quant aux méthodes non linéaires, il y a, entre autres, les concepts d'exposant de Lyapunov [66] qui représente le taux exponentiel moyen de divergence ou convergence d'orbites proches dans l'espace de phase. Une valeur positive de cet exposant indique que le comportement est chaotique. À l'inverse, un comportement stable est caractérisé par un exposant négatif ; de dimensions de corrélation, i.e. une des nombreuses estimation de la dimension fractale [51] ; d'entropie [51]. Cette notion caractérise le degré de désorganisation ou le manque d'information d'un système. Cependant, bien qu'il s'agisse d'un domaine très étudié et mature, l'analyse des séries temporelles présente certaines limites, en particulier lorsqu'il s'agit d'étudier des systèmes complexes.

Depuis quelques années, plusieurs auteurs se sont intéressés à une nouvelle façon d'analyser les séries temporelles en les transformant en un autre objet mathématique que sont les réseaux. Cette idée est attrayante puisqu'elle permet de lier deux domaines, l'analyse des signaux et la théorie des graphes. De nombreuses stratégies ont été présentées. À titre d'exemple, les auteurs de [69] ont développé une méthode faisant correspondre chaque cycle d'un signal pseudo-périodique à un nœud. Ils ont montré que les signaux périodiques "bruités" correspondent à des réseaux aléatoires tandis que des signaux chaotiques génèrent des réseaux qui présentent des propriétés *small world* et *scale free*. Dans [36], les auteurs proposent une méthode basée sur les propriétés de récurrence dans l'espace de phase d'un système dynamique. En particulier, la matrice de récurrence [17] obtenue en imposant un seuil dans la distance minimale entre deux points dans l'espace de phase (un peu à l'instar de [67]) est interprétée comme étant la matrice d'adjacence d'un graphe non dirigé et non pondéré. Cette méthode a été utilisée pour différencier les séries stochastiques des séries chaotiques. Dans ce travail, nous expliquons deux autres méthodes telles que développées dans [11, 41].

Comme mentionné précédemment, étudier les interactions entre les composants d'un système permet également de comprendre l'organisation de celui-ci. Cependant, pour un système réel, il est souvent difficile de détecter ces interactions par des méthodes physiques à cause du grand nombre de composants. Pour cela, il est courant de considérer les composants comme étant des nœuds d'un réseau et les interactions sous-jacentes comme les liens entre les nœuds. Dès lors, étudier la connectivité du réseau permet de comprendre l'organisation du système en question. Il existe de nombreuses méthodes permettant l'inférence de réseau. À titre d'exemple, il y a les méthodes basées sur la cross-corrélation, l'information mutuelle, *Granger causality* [47, 10, 70]. Dans ce travail, nous expliquons deux méthodes basées sur la théorie de l'information [8, 55].

Ce travail s'articule en six chapitres. Dans le chapitre 1, nous rappelons quelques définitions de la théorie de graphes et exposons les outils de représentation des réseaux de ce travail. Les chapitres 2 et 3 proposent chacun une application pour faire correspondre un réseau à une série temporelle. La première méthode possède également une opération pseudo-inverse. Ces méthodes permettent d'analyser les séries temporelles sous un angle différent ; certaines informations contenues dans les séries temporelles de base se retrouvent dans les réseaux associés. Quant aux chapitres 4 et 5, ceux-ci proposent deux méthodes pour étudier les interactions entre les composants d'un système. En particulier, les composants sont des séries temporelles et nous essayons de décrire l'organisation d'un système sur base de l'information partagée entre ses composants. Enfin, le chapitre 6 permet d'appliquer les méthodes décrites dans les chapitres précédents sur des données réelles.

Chapitre 1

Rappels de théorie de graphes et représentation

Ce premier chapitre a pour objectif de rappeler certaines notions très basiques de la théorie des graphes ainsi que d'expliquer l'outil utilisé pour représenter certains graphes présents dans ce travail.

1.1 Définitions

Nous donnons quelques définitions relatives aux graphes. Ces définitions sont tirées de [31].

- Un graphe est un triplet (V, E, ψ) tel que
 - V est un ensemble de nœuds ou sommets ;
 - E est un ensemble d'arêtes ;
 - ψ est une fonction, dite fonction d'incidence, qui à chaque arête associe un sommet ou une paire de sommets. Toutefois, par la suite, nous écrirons uniquement (V, E) ;
- Un graphe dirigé, ou digraphe, est un triplet (V, E, ψ) tel que
 - V est l'ensemble des sommets ;
 - E est l'ensemble des arêtes ;
 - ψ est la fonction d'incidence, qui associe à chaque arête une paire ordonnée de sommets. Toutefois, par la suite, nous écrirons uniquement (V, E) ;
- Deux nœuds sont adjacents s'ils sont liés par une même arête ;
- Une boucle est un arête incidente à un seul sommet ;
- Le degré d'un nœud est le nombre d'arêtes incidentes à celui-ci ; une boucle comptant pour deux ;
- Un parcours est une suite $v_0 e_0 v_1 e_1 \dots v_n$ où les v_i sont des sommets et e_i des arêtes ;
- Un chemin est un parcours dont tous les sommets sont distincts ;
- Un graphe pondéré est un graphe (V, E, ψ) possédant une fonction de poids $w : E \rightarrow \mathbb{R}$ qui associe un nombre réel à chaque arête ;
- La matrice d'adjacence A d'un graphe est une matrice carrée $\gamma \times \gamma$ dont l'élément a_{ij} est le nombre d'arêtes entre le sommet i et le sommet j et où γ est le nombre de sommets du graphe ;
- Le diamètre d'un graphe représente la distance maximale entre chaque paire de nœuds.

1.2 Outil de représentation

Certains graphes présents dans ce travail sont représentés grâce au logiciel Gephi. Il s'agit d'une plateforme interactive de visualisation et d'exploration des réseaux et de systèmes complexes [7]. Ce logiciel a été utilisé car il permet de visualiser très simplement certaines caractéristiques d'un réseau. Nous n'en avons utilisé que deux, que nous détaillons ci-dessous.

Modularité

La modularité est un indice de qualité d'un partitionnement des nœuds d'un graphe en communautés, i.e. en ensembles de nœuds tels que chacun de ces ensembles soit densément connecté en interne [40]. La modularité prend ses valeurs entre -1 et 1. Lorsqu'elle vaut -1, cela signifie que toutes les arêtes sont intercommunautaires. À l'inverse, lorsque les arêtes sont toutes intracommunautaires, la modularité vaut 1. La modularité tend vers 0 lorsque les arêtes d'un graphe sont

distribuées aléatoirement. Au-delà de 0.3, nous pouvons considérer que le graphe admet une structure de communauté significative. D'après Newman [40], les valeurs de modularités pour des graphes admettant des communautés s'étendent de 0.3 à 0.7. Des valeurs supérieures à 0.7 sont très peu fréquentes. De façon formelle, la modularité Q s'exprime

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) \delta(c_i, c_j) \quad (1.1)$$

où

- m est le nombre d'arêtes du graphe et vaut $\frac{1}{2} \sum_{i,j} A_{i,j}$;
- A est la matrice d'adjacence pondérée du graphe ;
- k_i est la somme des poids des arêtes du nœud i ;
- δ est la fonction delta de Kronecker ;
- c_i est la communauté à laquelle le nœud i appartient.

L'indice de modularité mesure donc la différence entre la densité d'un graphe donné et la densité d'un graphe aléatoire possédant les mêmes propriétés (nombre et poids de liens).

La méthode utilisée dans ce travail pour calculer la modularité d'un réseau est la méthode dite de Louvain [9]. À l'instar de beaucoup d'autres algorithmes de clustering, la qualité du regroupement en communautés est optimisée par la maximisation de la fonction de modularité (équ. (1.1)). Maximiser cette fonction pour diviser un graphe équivaut à vouloir que le nombre et poids des liens soient plus importants à l'intérieur des communautés qu'en dehors. La méthode de Louvain fonctionne de façon itérative. Au départ, tous les nœuds appartiennent à des partitions différentes ; il y a donc autant de communautés que de nœuds. Ensuite, pour chaque nœud i , nous considérons tous ses voisins j et nous évaluons le gain que nous aurions si ce nœud i faisait partie d'une communauté de j . Le nœud i est placé dans la communauté maximisant le gain, à condition que celui-ci soit positif. Si ce n'est pas le cas, nous laissons le nœud i où il est. Cette procédure est appliquée sur chaque nœud jusqu'à ce qu'il n'y ait plus moyen d'améliorer la modularité ; nous avons donc atteint des partitions de modularité maximale locale.

Il est possible de montrer que le gain de modularité ΔQ obtenu lorsque nous déplaçons un nœud isolé i dans une communauté C vaut

$$\Delta Q = \left(\frac{\sum_{in} + k_{i,in}}{2m} + \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right) - \left(\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right) \quad (1.2)$$

avec

- \sum_{in} est la somme des poids des liens dans la communauté C ;
- \sum_{tot} est la somme des poids des liens incidents aux nœuds de C ;
- k_i est la somme des poids des liens incidents au nœud i ;
- $k_{i,in}$ est la somme des poids des liens partant du nœud i vers les autres nœuds de C ;
- m est la somme de tous les poids de tous les nœuds du graphe.

Une expression semblable à (1.2) est utilisée lorsque nous retirons le nœud i de sa communauté.

Ensuite, nous reconstruisons un réseau ayant cette fois pour nœuds les communautés déterminées précédemment. La somme des poids des liens des nœuds entre deux communautés nous fournit le poids du lien entre les nouveaux nœuds du nouveau graphe. Une fois le nouveau graphe reformé, nous appliquons à nouveau la procédure faite pour les "nœuds isolés" et ainsi de suite jusqu'à ce qu'il n'y ait plus moyen d'améliorer la valeur de la modularité. La figure 1.1 illustre cet algorithme.

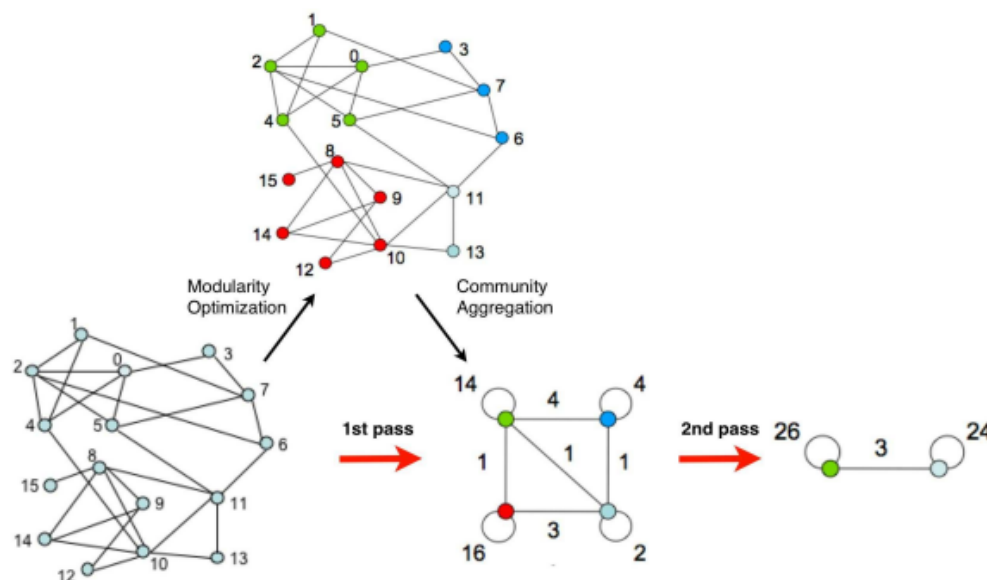


FIGURE 1.1 – Visualisation de l'algorithme de Louvain [9].

Cette méthode a l'avantage de procéder "bottom-up" et donc de ne pas tuer des petites communautés ayant un bon score de modularité dans une plus grande structure moins significative. De plus, elle est relativement rapide ; sa complexité en temps d'exécution est en $O(n \log n)$. Cela est dû au fait que les gains en modularité se calculent facilement (éq (1.2)) et que le nombre de communautés décroît à mesure que nous avançons dans l'algorithme. Par contre, elle présente un petit inconvénient : l'ordre dans lequel les nœuds sont considérés influence le résultat de la modularité. Différentes propositions sont envisagées pour contrer ce problème. La première est d'effectuer des permutations pour s'assurer de la validité du résultat. Une seconde option est d'ordonner les nœuds en fonction des caractéristiques du réseau.

Spatialisation

Il existe de nombreuses possibilités de spatialisation. Nous utilisons l'une où l'autre en fonction des caractéristiques que nous désirons accentuer. Dans la suite, nous utiliserons *OpenOrd*. Créé par Martin, Brown, Klavaus et Bayack, son but est de mieux distinguer les différentes communautés. Il a l'avantage d'être très rapide. Sa complexité est en $O(N \log(N))$ et peut être appliqué pour de graphes de 100 à 1 000 000 nœuds.

L'origine de cet algorithme est la méthode Fruchterman-Reingold [19]. Cet algorithme fonctionne avec un nombre d'itérations fixé et utilise un recuit simulé. En outre, il est composé de cinq étapes différentes : liquide, expansion, refroidissement, rupture et mijotement (*liquid*, *expansion*, *cool-down*, *crunch* et *simmer*). Les paramètres tels que la température, l'attraction ou encore l'amortissement changent d'une étape à l'autre. Les itérations sont partagées entre ces étapes et la distribution du temps se fait, *grosso modo*, comme suit : 25% de son temps en phase liquide, 25% en phase d'expansion, 25% en phase de refroidissement, 10% en phase de rupture et les 15% restant au stade de mijotement. Les paramètres sur lesquels nous pouvons directement (via Gephi) jouer sont

- edge cut : allant de 0 (Fruchterman-Reingold standard) à 1, cela représente les plus grandes distances entre deux nœuds. Une grande valeur indique un résultat plus "clusterisé" ;
- num iteration : permet de contracter ou d'étendre les clusters ;
- random seed.

En résumé, cet algorithme est capable de calculer des communautés (bien que pour une représentation visuelle couleur ainsi que pour les valeurs numériques, nous utilisons la méthode de Louvain) et d'agencer les réseaux par communautés ; nous l'utilisons uniquement pour son aspect "spatial" et non le calcul en soi des communautés.

Chapitre 2

Méthode des quantiles

L'objectif de ce second chapitre est de présenter une méthode permettant de transformer une série temporelle en un graphe et inversement. Cette méthode est basée sur un découpage du domaine de la série temporelle. L'avantage de cette méthode est de proposer une application possédant un pseudo inverse défini de façon naturelle, qui repose sur le concept de marche aléatoire sur un réseau. Ce chapitre se base sur l'article [11].

2.1 Aspects théoriques

Soit \mathcal{M} une application envoyant une série temporelle (à valeurs) continue(s) $X = \{x(t) | t \in \mathbb{N}, x(t) \in \mathbb{R}\} \in \mathcal{T}$, avec \mathcal{T} le domaine des séries temporelles, dans un réseau $g = \{\mathcal{N}, \mathcal{A}\} \in \mathcal{G}$, avec \mathcal{G} le domaine des graphes, où \mathcal{N} et \mathcal{A} sont un ensemble de nœuds et d'arêtes respectivement. L'idéal serait que cette fonction retienne toute l'information de la série d'origine, éventuellement par l'existence d'une fonction bijective \mathcal{M}_b où l'unique image d'une série X serait $g = \mathcal{M}_b[X]$ et telle que $X = \mathcal{M}_b^{-1}[\mathcal{M}_b[X]]$. Malheureusement, cela n'est pas faisable en pratique étant donné que X est continue et qu'un graphe ne contient jamais qu'un nombre fini de nœuds. Une façon de contrer ce problème est de discrétiser le domaine de la série. Pour ce faire, nous définissons ses \mathcal{Q} quantiles et faisons correspondre à chaque quantile q_i un nœud $n_i \in \mathcal{N}$ dans le graphe correspondant. Pour rappel, le quantile q_p d'ordre p ($0 < p < 1$) d'une série $\{x_i\}_{i=1,\dots,n}$ donnant lieu à la série ordonnée $\{x_{(i)}\}_{i=1,\dots,n}$ correspond à la donnée de rang $[np]$, $x_{([np])}$, où $[np]$ est l'arrondi vers le haut de np . La probabilité qu'un point dans le quantile q_i précède un nœud dans le quantile q_j représente le poids w_{ij} de l'arête connectant les nœuds n_i et n_j (Figure 2.1).

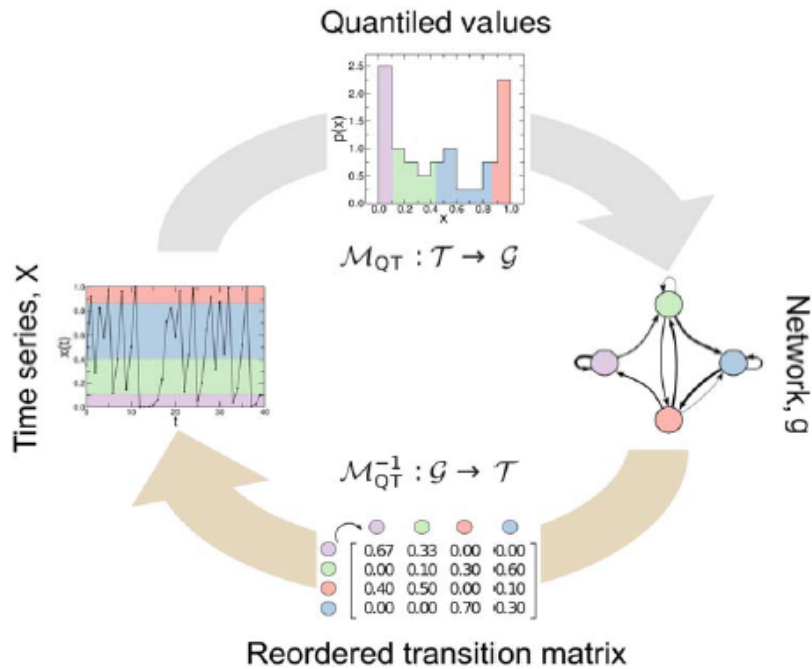


FIGURE 2.1 – Illustration de la méthode : le domaine de la série temporelle est divisé en quatre quantiles représentant chacun un nœud dans le graphe associé [11].

 Algorithme de \mathcal{M}_{QT}

1. $q \leftarrow$ vecteur de \mathcal{Q} quantiles
 2. for $i = 1$: nombre de points
 3. $l \leftarrow$ quantile de $x(i)$
 4. $j \leftarrow$ quantile de $x(i+1)$
 5. $W(l, j) = W(l, j) + 1$
-

Par exemple, la série temporelle X de la figure 2.1, reprise à la figure 2.2, est divisée en $\mathcal{Q} = 4$ quantiles (colorés). Cette série est composée de $N = 40$ données et pour déterminer les quantiles, nous ordonnons par ordre croissant les données. Dans notre exemple, la valeur de la $N/Q \times 1 = 40/4 = 10$ ème donnée nous donne la limite du premier quantile. Le deuxième quantile sera délimité par les $N/Q \times 1 = 10$ ème et $N/Q \times 2 = 20$ ème données de la série ordonnée et ainsi de suite, jusqu'au quatrième quantile. De façon générale, le k -ème quantile est délimité par les données $N/Q \times (k-1)$ et $N/Q \times k$ de la série ordonnée. Chaque quantile q_i est associé à un nœud n_i dans le graphe g correspondant. Pour le calcul des probabilités, i.e. pour obtenir la matrice de poids, prenons par exemple le quantile vert. Pour chaque point $x(t)$ dans ce quantile, nous regardons le quantile de $x(t+1)$.

$x(t)$ du quantile vert	$x(0)$	$x(2)$	$x(6)$	$x(7)$	$x(9)$	$x(17)$	$x(22)$	$x(26)$	$x(29)$	$x(34)$
Quantile de $x(t+1)$	Rouge	Bleu	Vert	Rouge	Bleu	Bleu	Bleu	Bleu	Rouge	Bleu

En résumé, nous avons 1 Vert, 3 Rouge et 6 Bleu. Pour avoir les chiffres de la matrice de la figure 2.2, nous les prenons en probabilités, i.e. entre 0 et 1. Puisque nous avons dix données dans ce quantile, nous divisons pas dix les chiffres obtenus. Nous avons donc bien Vert \rightarrow Vert : 0.1, Vert \rightarrow Rouge : 0.3 et Vert \rightarrow Bleu : 0.6. Il en va de même pour les autres quantiles.

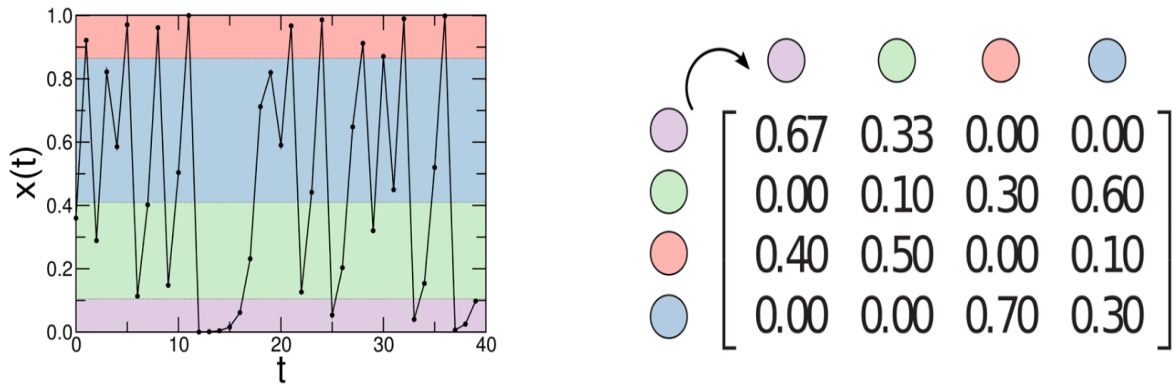


FIGURE 2.2 – Illustration de la méthode - calcul des probabilités [11].

La fonction proposée, dénotée par \mathcal{M}_{QT} possède trois propriétés importantes. Premièrement, elle est surjective c'est-à-dire qu'une série X avec $t = 1, 2, \dots, T$ ne produit qu'un et un seul réseau $g = \mathcal{M}_{QT}[X]$. Notons tout de même qu'un graphe g peut être obtenu à partir de deux séries distinctes X et X' : $g = \mathcal{M}_{QT}[X] = \mathcal{M}_{QT}[X']$. Deuxièmement, lorsque $T > \mathcal{Q}$, le graphe résultant est pondéré, dirigé et connexe. Enfin, \mathcal{M}_{QT} requiert uniquement la connaissance de \mathcal{Q} .

Cette application à l'avantage de posséder un pseudo-inverse : une réalisation d'une marche aléatoire sur le réseau avec les probabilités de transition w_{ij} données par la matrice adjacente des poids telle que $\sum_j w_{ij} = 1$ (figure 2.1). Pour retrouver (en partie) la série d'origine, nous commençons à partir d'un nœud du graphe pris au hasard et effectuons une marche aléatoire sur le réseau. La probabilité d'arriver sur le nœud n_j en partant du nœud n_i est donnée par w_{ij} . En identifiant chaque nœud à un quantile, nous construisons la série en choisissant à chaque pas de la marche aléatoire une valeur aléatoire suivant une loi uniforme dans le quantile correspondant.

Algorithme de \mathcal{M}_{QT}^{-1}
1. $n_i \leftarrow$ nœud départ
2. $q \leftarrow$ quantile de n_i
3. $x(1) \leftarrow$ valeur arbitraire dans q
4. $k \leftarrow 2$
5. while (encore des nœuds)
6. $n_j \leftarrow$ nœud sur lequel va n_i avec probabilité w_{ij}
7. $q \leftarrow$ quantile de n_j
8. $x(k) \leftarrow$ valeur arbitraire dans q
9. $k \leftarrow k + 1$
10. $n_i \leftarrow n_j$

En l'absence d'une connaissance *a priori* d'une correspondance directe entre les quantiles et les nœuds, nous supposons la série temporelle aussi lisse que possible. Pour ce faire, nous réorganisons la matrice d'adjacence W de telle façon que les poids les plus importants se situent près de la diagonale. Pour trouver une telle disposition de matrice, nous utilisons la fonction de poids (2.1) où chaque élément de la matrice est "relié" à la diagonale en fonction de son poids

$$C = \frac{1}{N} \sum_{i,j=1}^N w_{ij} |i - j| \quad (2.1)$$

où N est la dimension de la matrice des probabilités de transition. Le but est de minimiser cette fonction. Pour ce faire, nous choisissons aléatoirement à chaque itération une paire de nœuds et échangeons leur position. Les autres nœuds ne bougent pas. Enfin, nous évaluons la fonction (2.1) avec la nouvelle matrice et gardons la disposition qui minimise la fonction sur k itérations, k fixé. Par conséquent, la probabilité de faire des "petits sauts" est augmentée. En effet, supposons que nous soyons sur le nœud k . Nous serons dès lors plus susceptibles d'aller sur le nœud $k \pm n$ avec n un petit entier que d'aller sur le nœud $k \pm f$ avec f un grand entier. La série résultante sera donc plus lisse puisque nous avons moins de probabilités de sauter d'un quantile à un autre fort éloigné.

Algorithme "arrangement matrice"
1. $cost \leftarrow$ évaluation eq (2.1) pour W
2. while (non condition d'arrêt)
3. $n_1, n_2 \leftarrow$ nœuds à échanger pris aléatoirement
4. $W_{n_1, n_2} \leftarrow W$ avec n_1, n_2 permutés
5. $new_cost \leftarrow$ évaluation eq (2.1) pour W_{n_1, n_2}
6. if $new_cost < cost$
7. $W \leftarrow W_{n_1, n_2}$
8. $cost \leftarrow new_cost$

Reprenons l'exemple des figures 2.1 et 2.2. Si nous ne connaissons pas la relation entre les nœuds et les quantiles, nous réorganisons la matrice d'adjacence "de base" (celle obtenue directement lorsque nous construisons le réseau) par l'algorithme expliqué ci-dessus ce qui nous permet d'obtenir la matrice indiquée sur les figures 2.1 et 2.2. Pour cet exemple, le gain du coût est relativement faible. Il n'y a pas vraiment de gain "attendu" ; cela est évidemment fonction du signal de base ainsi que du nombre de quantiles considéré.

$$\underbrace{\begin{pmatrix} 0.67 & 0.33 & 0.00 & 0.00 \\ 0.00 & 0.10 & 0.60 & 0.30 \\ 0.00 & 0.00 & 0.30 & 0.70 \\ 0.40 & 0.50 & 0.10 & 0.00 \end{pmatrix}}_{\substack{\text{Matrice "de base"} \\ \text{Fonction poids : } C = 1.13}} \xrightarrow{\text{arrangement}} \underbrace{\begin{pmatrix} 0.67 & 0.33 & 0.00 & 0.00 \\ 0.00 & 0.10 & 0.30 & 0.60 \\ 0.40 & 0.50 & 0.00 & 0.10 \\ 0.00 & 0.00 & 0.70 & 0.30 \end{pmatrix}}_{\substack{\text{Matrice réordonnée} \\ \text{Fonction poids : } C = 0.98}}$$

Tout comme \mathcal{M}_{QT} , cette application notée \mathcal{M}_{QT}^{-1} est également surjective. Étant donné un réseau g , la fonction produira une série $X = \mathcal{M}_{QT}^{-1}[g, \varepsilon]$ sur une réalisation ε des marches aléatoires. Notons que deux réseaux distincts g

et g' peuvent produire la même série $X = \mathcal{M}_{QT}^{-1}[g, \varepsilon] = \mathcal{M}_{QT}^{-1}[g', \varepsilon']$. Cependant, ce n'est pas réellement du *one-to-one* puisqu'il y a des éléments stochastiques qui interviennent. Néanmoins, bien que la carte proposée ne soit pas strictement *one-to-one*, un réseau donnera des séries temporelles très similaires malgré des réalisations différentes. Différentes mesures sont envisagées dans la suite pour quantifier cette dernière affirmation.

2.2 Quelques résultats

En vue de tester la validité de cette méthode, nous l'avons appliquée sur différentes séries temporelles.

2.2.1 Séries périodiques et aléatoires

Dans cette sous-section, nous avons considéré un ensemble de séries temporelles, périodiques comme aléatoires définies par

$$x(t) = \begin{cases} \text{mod}(x(t-1) + \delta + \eta, 1) & \text{avec probabilité } p \\ \text{mod}(x(t-1) + \delta, 1) & \text{sinon} \end{cases} \quad (2.2)$$

où δ est une constante, p est une probabilité en fonction de laquelle nous ajoutons un bruit η . η est une variable aléatoire suivant une loi uniforme $U([0,1])$. Nous avons généré les séries temporelles de $T = 320$ points pour différentes valeurs de p : $p = 0$; 0.01 ; 0.1 et 1. Nous avons également fixé δ à 0.05. Nous avons ensuite appliqué la fonction \mathcal{M}_{QT} avec $Q = 20$ quantiles. Ces séries et graphes sont appelés de la première génération. Nous remarquons que des séries avec des propriétés différentes sont envoyées sur des réseaux avec des propriétés différentes (figure 2.3).

Ensuite, nous avons appliqué la fonction inverse \mathcal{M}_{QT}^{-1} à chacun des graphes de la première génération pour obtenir une série dite de deuxième génération. Par soucis de simplicité, nous avons fait correspondre chaque quantile au quantile correspondant de la première génération de la série. Encore un fois, nous avons pris $T = 320$. Les séries de la première et de la deuxième génération sont relativement similaires comme le montrent les distances, les fonctions d'autocorrélation et le spectre de puissance entre les différentes générations dans la suite (figures 2.4 et 2.5).

Pour finir, nous avons de nouveau appliqué \mathcal{M}_{QT} aux séries de la deuxième génération pour obtenir des graphes de deuxième génération. Encore un fois, nous avons pris $Q = 20$. Nous remarquons également la similitude des graphes des différentes générations et ce, pour n'importe quelle valeur de p . Ces "similitudes" sont analysées dans la suite (figure 2.6).

Nous calculons la distance entre les séries de première et seconde génération. Au plus cette mesure est petite, au plus nos signaux prennent les mêmes valeurs au même moment. La distance utilisée [56] entre deux signaux x et y de longueur N s'exprime comme suit

$$d(x, y) = \left(\frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2 \right)^{1/2}. \quad (2.3)$$

La figure 2.4 reprend les distances calculées pour les différentes valeurs de p envisagées. Nous remarquons que pour 20 quantiles, nous obtenons toujours une très petite distance entre les signaux, contrairement aux autres valeurs de p pour lesquelles la distance tend à être de plus en plus élevée et varie d'une réalisation à l'autre.

Comme mentionné précédemment, nous avons comparé les fonctions d'autocorrélation [58] des différentes générations. L'autocorrélation discrète R d'une fonction x est la corrélation croisée de cette fonction avec elle-même à intervalle de temps j

$$R(j) = \sum_n (x_n - m)(x_{n-j} - m) \quad (2.4)$$

où m est la valeur moyenne attendue de x . Dans le cas continu, l'autocorrélation d'une fonction x (avec intervalle de temps τ) s'exprime

$$R_x(\tau) = x(\cdot - \tau)x(\cdot) = \int_{-\infty}^{+\infty} x(t + \tau)x(t)dt = \int_{-\infty}^{+\infty} x(t)x(t - \tau)dt. \quad (2.5)$$

L'autocorrélation est donc une représentation mathématique du degré de similarité entre une série temporelle et une version décalée d'elle-même. Il s'agit d'un cas particulier de la corrélation qui elle, est calculée entre deux séries temporelles différentes.

L'autocorrélation prend ses valeurs entre -1 et 1. Une valeur de 1 indique une parfaite corrélation positive : une augmentation observée dans une série conduit à une augmentation proportionnelle dans l'autre. À l'inverse, lorsque l'autocorrélation est de -1, cela signifie une parfaite corrélation négative : une augmentation dans une série conduit à une diminution dans l'autre.

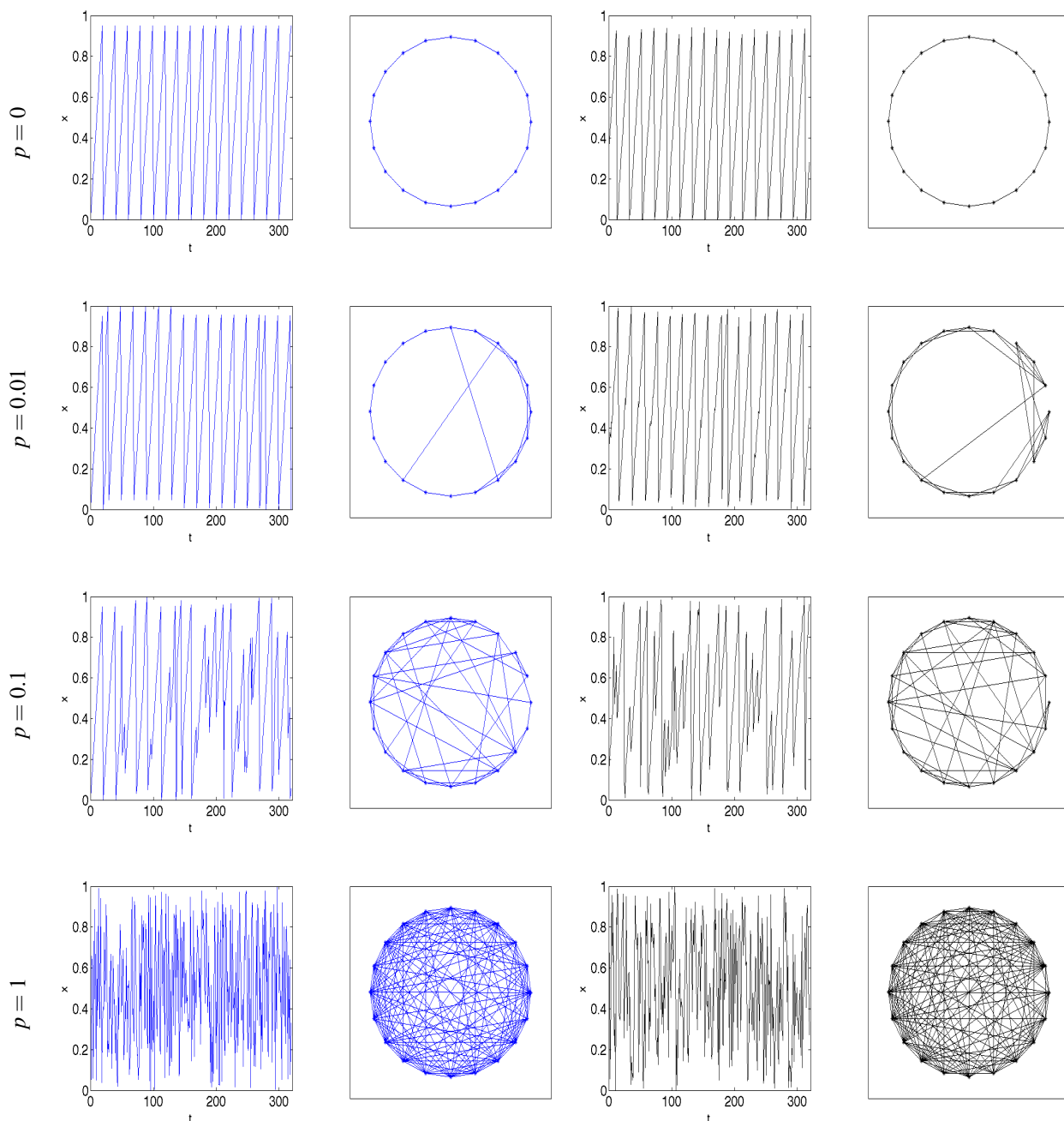


FIGURE 2.3 – Application des cartes \mathcal{M}_{QT} et \mathcal{M}_{QT}^{-1} sur des séries périodiques et aléatoires en considérant $\mathcal{Q} = 20$ quantiles. En bleu : première génération, en noir : seconde génération. La série est périodique lorsque $p = 0$ et au plus p se rapproche de 1, au plus les séries deviennent aléatoires. Les séries et graphes sont relativement similaires (voir analyses) entre les premières et secondes générations.

Nous observons (figure 2.5) que, quelle que soit la génération envisagée, l'autocorrélation révèle un signal distinct lorsque la série est périodique ($p = 0$) qui tend à disparaître au plus la série devient aléatoire ($p = 1$).

Nous nous sommes également intéressés au spectre de puissance. Ce dernier correspond à l'intensité du signal pour une fréquence donnée et est calculé comme étant le carré du module de la transformée de Fourier discrète du signal [65]. La transformée de Fourier discrète pour un signal x composé de N valeurs est

$$S_k = \sum_{n=0}^{N-1} x_n e^{-2i\pi k \frac{n}{N}}, \quad 0 \leq k < N. \quad (2.6)$$

Lorsque le signal est réel, nous avons que $S_k^* = S_{N-k}$. Puisque la transformée de Fourier est périodique et que les fréquences négatives sont donc égales aux positives, l'information à retenir se trouve entre les fréquences 0 et $\frac{N+P}{2}$ où P est la période.

La méthode pour calculer la transformée de Fourier discrète est la transformation de Fourier rapide (FFT). "Ra-

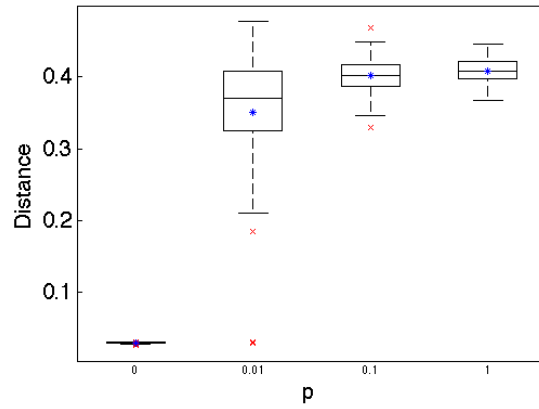


FIGURE 2.4 – Distance entre les séries de première et seconde générations pour $\mathcal{Q} = 20$ quantiles. Les résultats sont obtenus à partir de 100 réalisations. La croix bleue indique la moyenne pour chaque p considéré. Une valeur de $p = 0$ signifie que la série est périodique. Au plus p se rapproche de 1, au plus la série temporelle est aléatoire. Pour $p = 0$, nous obtenons toujours une très petite distance entre les signaux, contrairement aux autres valeurs de p pour lesquelles la distance tend à être de plus en plus élevée et varie d'une réalisation à l'autre.

vide" vient du fait que sa complexité est en $O(n \log(n))$ pour une série de n points, comparé à $O(n^2)$ pour l'algorithme "naïf". L'idée de base est "diviser pour régner" à l'aide d'une récurrence, c'est-à-dire que l'algorithme subdivise une transformation de Fourier discrète de taille $N = N_1 N_2$ en plusieurs transformées de tailles plus petites N_1 et N_2 (souvent $N/2 = N_1 = N_2$).

$$\begin{aligned}
 S_k &= \sum_{n=0}^{N-1} x_n e^{-2\pi i n k / N} \\
 &= \sum_{n=0, \text{pair}}^{N-1} x_n e^{-2\pi i n k / N} + \sum_{n=0, \text{impair}}^{N-1} x_n e^{-2\pi i n k / N} \\
 &= \sum_{n=0}^{N/2-1} x_{2n} e^{-2\pi i 2n k / N} + \sum_{n=0}^{N/2-1} x_{2n+1} e^{-2\pi i (2n+1) k / N} \\
 &= \sum_{n=0}^{N/2-1} x_{n,1} e^{-2\pi i n k / (N/2)} + e^{-2\pi i k / N} \sum_{n=0}^{N/2-1} x_{n,2} e^{-2\pi i n k / (N/2)} \\
 &= S_{k,1} + e^{-2\pi i k / N} S_{k,2}.
 \end{aligned} \tag{2.7}$$

D'où la DFT à N points peut être obtenue à partir de deux transformées à $N/2$ points ; l'une à partir des données paires, l'autre à partir des données impaires. Dès lors, lorsque N est une puissance de 2, il est possible d'appliquer récursivement cette décomposition jusqu'à ce qu'il ne reste plus que des transformées de Fourier d'un seul point.

Pour notre exemple, nous avons donc calculé la transformée de Fourier discrète au moyen de la FFT. Nous avons ensuite pris le carré des valeurs de la DFT et avons obtenu les résultats repris à la figure 2.5. Comme pour l'autocorrélation, nous avons des pics pour les séries périodiques et au plus les séries deviennent aléatoires, au plus ces pics s'amenuisent et le spectre de puissance se remplit d'autres "petits pics".

Si nous nous intéressons maintenant à la fonction inverse \mathcal{M}_{QT}^{-1} (figure 2.6), nous pouvons par exemple comparer le *In-Strength*. Nous remarquons que cette mesure vaut 1 pour tous les nœuds lorsque les réseaux sont réguliers ($p = 0$). Lorsque les réseaux tendent à être plus aléatoires, la distribution de *In-Strength* s'étend à cause de la redistribution des poids. Notons que le *Out-Strength* de chaque nœud est unitaire dans tous les cas puisque les poids sont des probabilités de Markov. Le poids des liens est unitaire lorsque le réseau est régulier et au plus le réseau devient aléatoire, au plus ces poids diminuent. Les plus courts chemins sont distribués de manière uniforme lorsque le réseau est régulier.

Nombre de quantiles

Nous faisons à présent varier le nombre de quantiles. De façon évidente, ce nombre doit varier entre 2 et le nombre de points composant le signal. De fait, il ne serait pas très intéressant de n'avoir qu'un seul nœud dans le réseau tandis qu'avoir plus de quantiles que de points produirait des réseaux ayant des nœuds isolés. Cela impliquerait que, pour reconstruire le signal, soit ces nœuds isolés ne seraient jamais atteints, soit nous resterions à jamais sur un de ces nœuds. Dans le premier cas, ces nœuds isolés seraient tout simplement inutiles, dans le second, cela induirait d'énormes erreurs.

Les résultats ci-dessus sont obtenus en considérant $\mathcal{Q}=20$. Au vu de la définition de x (eq. 2.2), nous voyons que

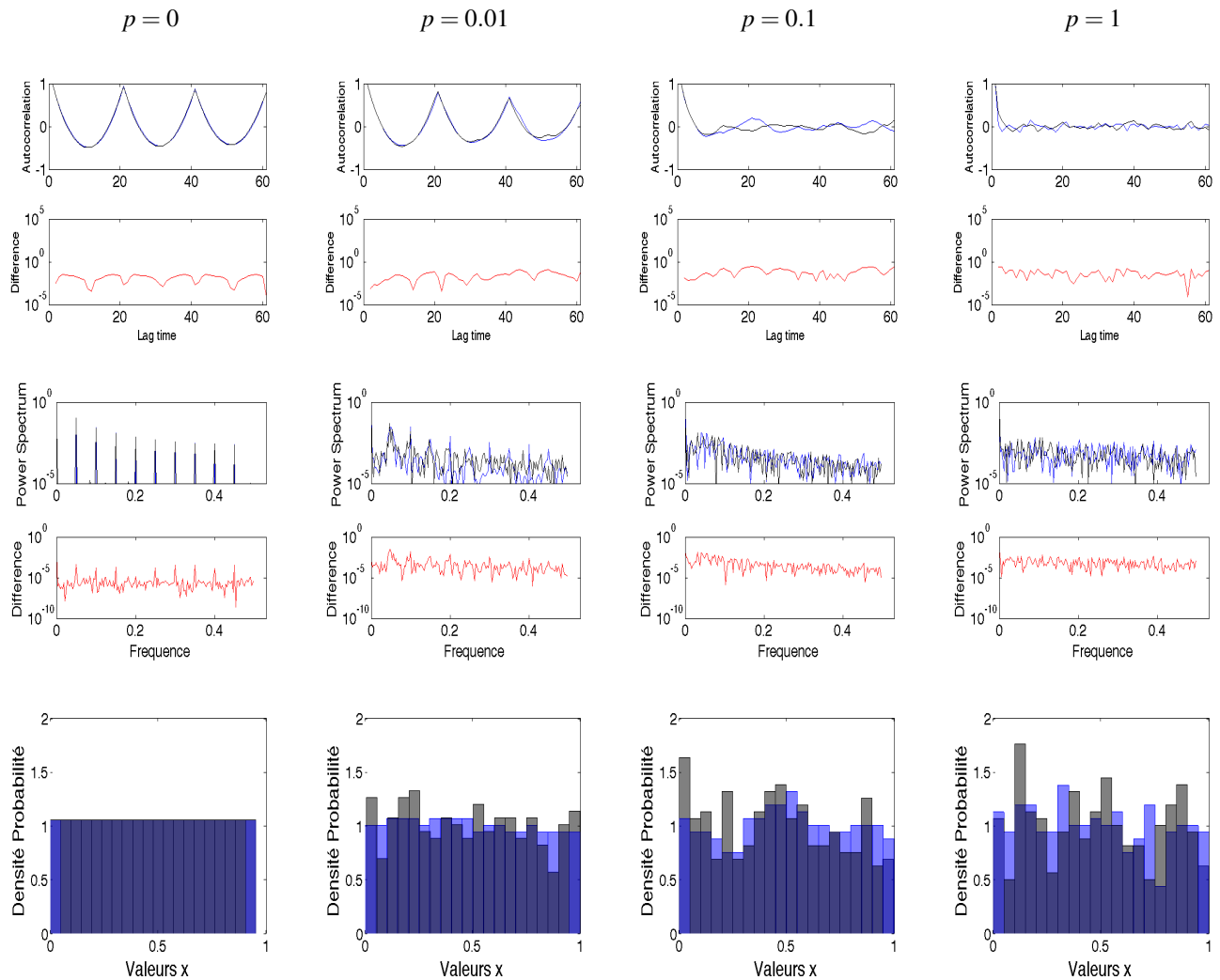


FIGURE 2.5 – Comparaison des propriétés statistiques entre la première et la seconde génération de séries temporelles en considérant $\mathcal{Q} = 20$ quantiles. En bleu : première génération ; en noir : seconde génération. Lorsque la série temporelle est périodique ($p = 0$), la fonction d'autocorrélation et le spectre de puissance révèlent un signal distinct, qui disparaît au fur et à mesure que p augmente. Lorsque $p = 0$, la probabilité de densité des valeurs est uniforme et au plus la série devient aléatoire, au plus cette uniformité s'amenuise.

$\mathcal{Q} = 20$ est le choix optimal lorsque $p = 0$. En effet, x ne prend que les valeurs 0, 0.05, 0.1, ..., 0.95 (et à chaque fois dans cet ordre). Nous avons donc exactement 20 valeurs différentes et lorsque nous considérons $\mathcal{Q} = 20$, chaque valeur correspond à un quantile. Dès lors, chaque quantile va toujours sur son suivant et lors de la reconstruction de la série de deuxième génération, tout est "déterminé". En effet, les seuls éléments non nuls de la matrice d'adjacence sont sur la diagonale juste au-dessus de la diagonale principale et prennent la valeur de '1' (pondéré). De plus, la distance euclidienne entre les deux signaux est quasi nulle, mais surtout bien plus petite que pour les autres quantiles (figure 2.8).

Lorsque nous diminuons le nombre de quantiles (par rapport à 20) nous remarquons qu'en ce qui concerne les séries temporelles, plus on se rapproche de 20, au plus la méthode semble adéquate pour la reconstruction. À l'inverse, lorsque le nombre de quantile augmente, nous ne remarquons pas d'amélioration ou de dégradation particulière (figures 2.7 et 2.8). Le fait d'avoir plus de quantiles que 20 a pour effet d'avoir des quantiles "vides", i.e. leurs bornes sont $[b_i, b_s]$ avec $b_i = b_s$ puisque nous n'avons que 20 valeurs différentes pour x . Par conséquent, nous envoyons des mêmes valeurs sur des nœuds différents et donc cela produit de moins bons résultats que $\mathcal{Q} = 20$. En ce qui concerne les réseaux, nous pouvons difficilement avancer une tendance autre que $\mathcal{Q} = 20$ permet de retrouver les mêmes caractéristiques.

Intéressons-nous à présent à la série aléatoire, i.e. lorsque $p = 1$. Cette fois, la valeur $\mathcal{Q} = 20$ n'a plus la même signification puisque la série peut prendre un nombre infini de valeurs. Au niveau des séries, nous remarquons sans surprise des allures similaires d'autocorrélation et de spectre de puissance. Par contre, il semble difficile de tirer une conclusion générale. La distance entre les deux signaux n'apporte pas plus d'information (figure 2.8) ; il semblerait que le nombre de quantiles n'intervienne pas lorsque la série est complètement aléatoire.

Quelques mesures sont reprises à la figure 2.7. Pour la modularité des réseaux, au plus le réseaux est aléatoire, au plus la modularité est petite. Une information importante se trouve à $\mathcal{Q} = 20$. C'est pour cette valeur que les différences entre première et seconde générations sont les plus petites. En particulier pour $p = 0$, nous avons une différence nulle, les modularités étant égales s'élèvent à 0.55. C'est le seul endroit où nous avons une différence nulle. Remarquons

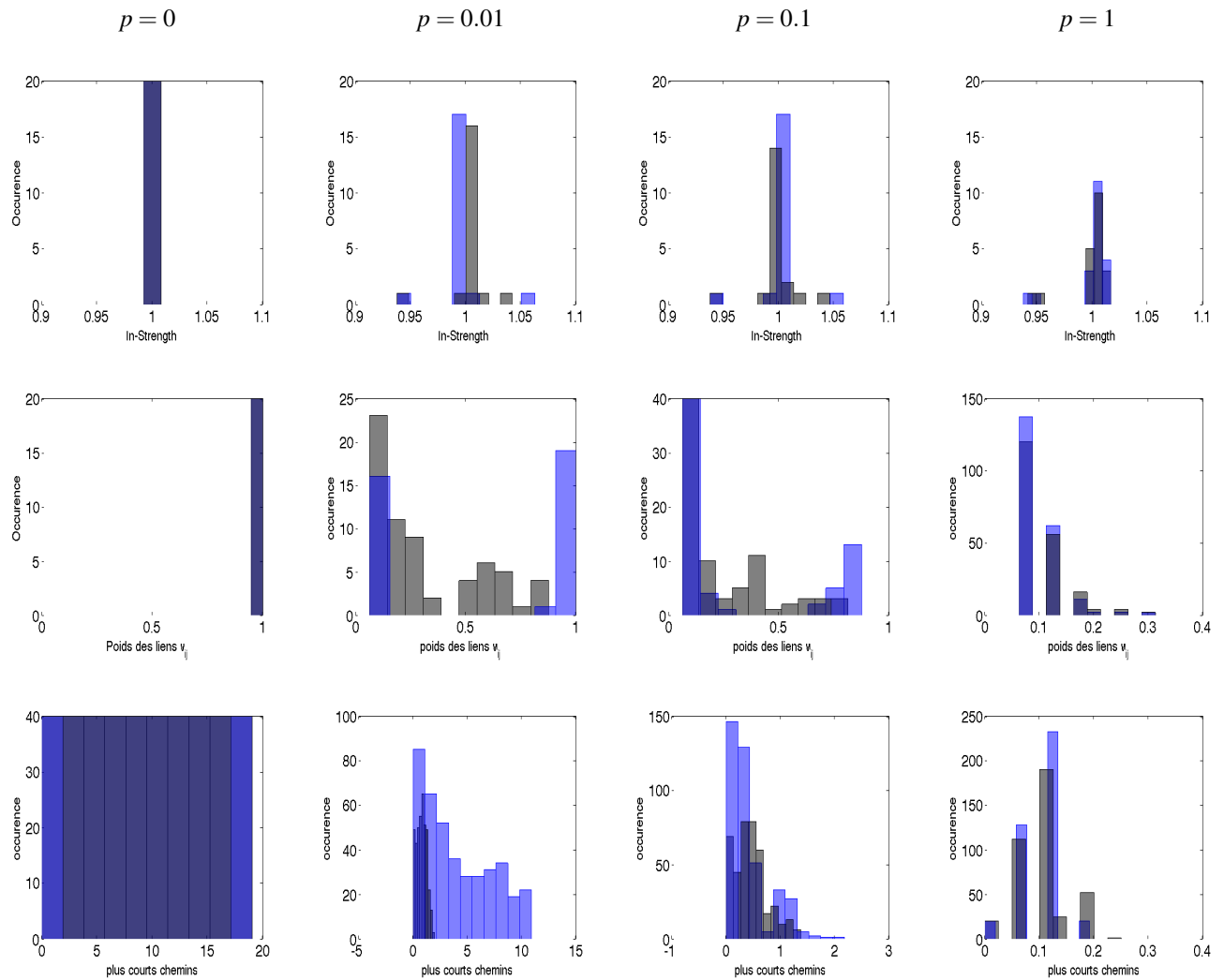


FIGURE 2.6 – Comparaison des propriétés statistiques entre la première et la seconde génération de réseaux en considérant $\mathcal{Q} = 20$ quantiles. Les valeurs de p situées en tête de colonnes fait référence à la série à laquelle le réseau est associé ; $p = 0$ correspond à une série périodique tandis que $p = 1$ correspond à une série aléatoire. En bleu : première génération ; en noir : seconde génération. *In-Strength* vaut 1 pour tous les nœuds lorsque les réseaux sont réguliers ($p = 0$). Lorsque les réseaux tendent à être plus aléatoires, la distribution de *In-Strength* s'étend à cause de la redistribution des poids. Notons que le *Out-Strength* de chaque nœud est unitaire dans tous les cas puisque les poids sont des probabilités de Markov. Le poids des liens est unitaire lorsque le réseau est régulier et au plus le réseau devient aléatoire, au plus ces poids diminuent. Les plus courts chemins sont distribués de manière uniforme lorsque le réseau est régulier. Au plus les réseaux deviennent aléatoires (p croît), au plus la distribution des plus courts chemins est en "pics".

également que la modularité des premières générations est toujours supérieure à celle de la deuxième. Au niveau des boucles, l'unique endroit où nous avons une différence entre première et seconde génération égale à 0 est lorsque $\mathcal{Q} = 20$ et $p = 0$. L'écart le plus important se trouve pour 32 quantiles, et ce, pour n'importe quel p considéré. Dans l'ensemble, c'est pour $p = 0$ que nous avons le moins de différences. Enfin, nous terminons avec le diamètre des graphes. Pour ce dernier, nous n'arrivons jamais à retrouver le même résultat pour quelque valeur que ce soit. Néanmoins, encore une fois, $\mathcal{Q} = 20$ et $p = 0$ nous donne la meilleur reconstruction du réseau.

2.2.2 Rossler & Lorenz

Nous appliquons \mathcal{M}_{QT} et \mathcal{M}_{QT}^{-1} sur deux séries temporelles différentes. En particulier, les séries sont composées de la variable x du système de Rossler ainsi que du système de Lorenz. Les équations de Rossler s'expriment comme

$$\begin{aligned} dx/dt &= -y - z \\ dy/dt &= x + ay \\ dz/dt &= b + z(x - c) \end{aligned}$$

avec les paramètres $a = 0.1$, $b = 0.1$ et $c = 14$. Ces équations différentielles définissent un système dynamique continu et tridimensionnel qui présente des caractéristiques chaotiques. Son espace de phase génère un attracteur chaotique avec un lobe. Une orbite dans l'attracteur suit une spirale proche du plan (x, y) autour d'un point fixe instable. Après que l'orbite ait décrit quelques spirales, un second point fixe provoque une élévation de cette orbite dans la direction z et ensuite une

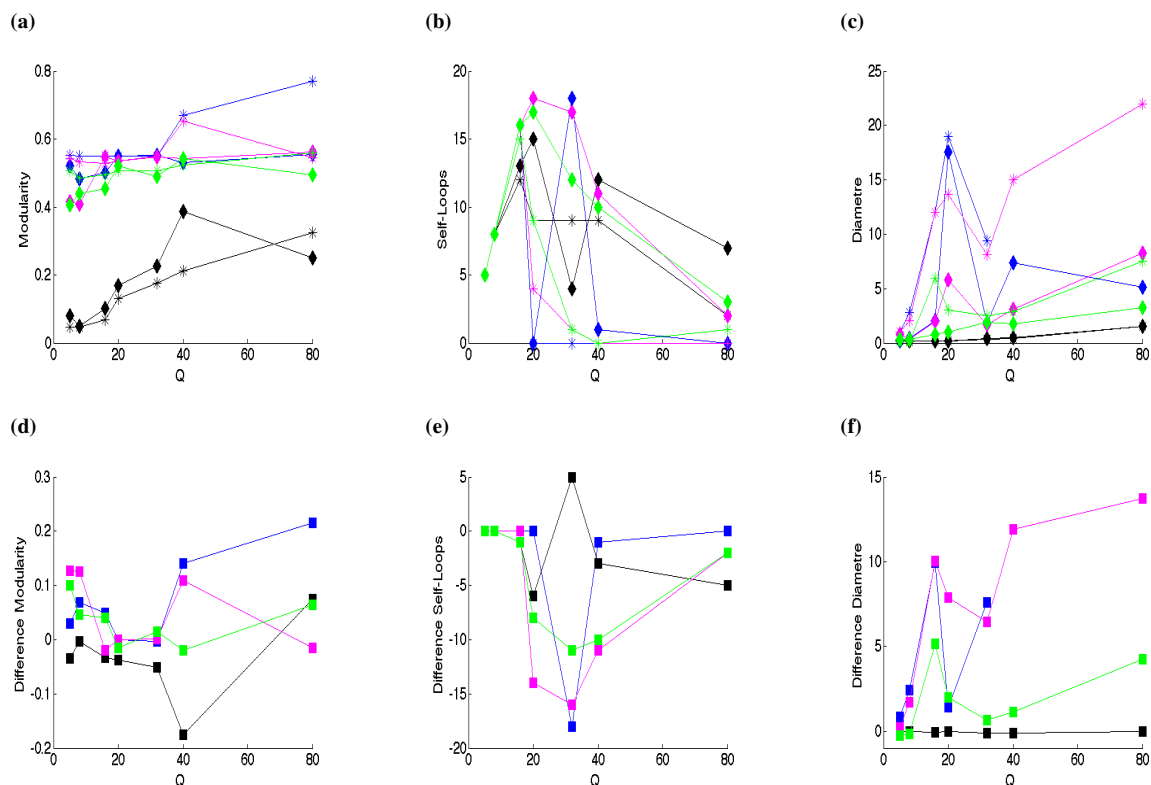


FIGURE 2.7 – (a) Modularité ; (b) Self-Loops ; (c) Diametre des différents réseaux obtenus en considérant différentes valeurs de p et de \mathcal{Q} . Bleu : $p = 0$, Magenta : $p = 0.01$, Vert : $p = 0.1$ et Noir : $p = 1$. Les valeurs de réseaux de première génération sont indiquées par des croix tandis que pour les secondes générations, ce sont des losanges. Les différences entre première et seconde générations des différentes mesures sont tracées sur les panels (d) à (f) respectivement. Seules sont représentées les moyennes, obtenues à partir de 10 réalisations. Modularité : au plus le réseau est aléatoire, au plus la modularité est petite. Pour $\mathcal{Q} = 20$, les différences entre première et seconde générations sont les plus petites. En particulier pour $p = 0$, la différence est nulle, les modularités étant égales s'élèvent à 0.55. Notons que la modularité des premières générations est toujours supérieure à celle de la deuxième. Loops : l'unique endroit où nous avons une différence entre première et seconde génération égale à 0 est lorsque $\mathcal{Q} = 20$ et $p = 0$. L'écart le plus important se trouve pour 32 quantiles, et ce, pour n'importe quel p considéré. Dans l'ensemble, c'est pour $p = 0$ que nous avons le moins de différences. Diametre : nous n'arrivons jamais à retrouver le même résultat pour quelque valeur que ce soit. Néanmoins, encore une fois, $\mathcal{Q} = 20$ et $p = 0$ nous donne la meilleur reconstruction du réseau.

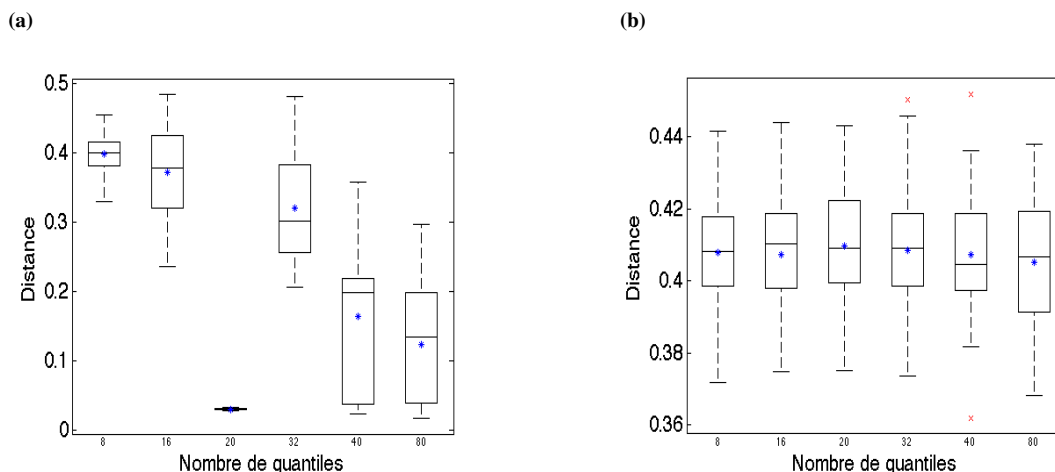


FIGURE 2.8 – Distances entre les séries (a) périodiques $p = 0$; (b) aléatoires $p = 1$ de première et deuxième génération en fonction du nombre d quantiles. La série périodique possède un nombre optimal de quantiles $\mathcal{Q} = 20$ puisque pour cette valeur, la série x ne prend que les valeurs 0, 0.05, 0.1, ..., 0.95 (et à chaque fois dans cet ordre). Il y a donc exactement 20 valeurs différentes et chaque valeur correspond à un quantile. La distance entre les deux générations est dès lors très faible et ne varie que très peu d'une réalisation à l'autre. Pour les autres valeurs de \mathcal{Q} , la valeur moyenne (ou médiane) est beaucoup plus élevée et les variations sont beaucoup plus marquées d'une réalisation à l'autre. La série aléatoire semble insensible à \mathcal{Q} avec des moyennes, médianes et variations presque identiques quel que soit \mathcal{Q} .

descente vers le plan (x, y) proche du premier point fixe, réintégrant l'orbite dans la spirale. Ce comportement génère un mouvement d'oscillations quasi périodique dans la variable x [49] (figure 2.12). Les équations de Lorenz sont

$$\begin{aligned} dx/dt &= \sigma(-x + y) \\ dy/dt &= rx - y - xz \\ dz/dt &= bz + xy \end{aligned}$$

avec les paramètres $\sigma = 10$, $r = 28$ et $b = 8/3$. Les solutions de ce système mènent à un attracteur dans un espace à trois dimensions (x, y, z) . Contrairement à Rossler, cet attracteur est "composé" de deux lobes. Pour presque toutes les conditions initiales (différentes de celles des points fixes), les orbites tournent autour d'un des deux points fixes et puis s'échappent de ce lobe pour aller s'enrouler sur l'autre lobe autour du second point fixe et ainsi que suite. Ce comportement est reconnaissable dans la variable x qui prend tour à tour des valeurs positives et négatives [32] (figure 2.13).

Pour appliquer la méthode proposée sur les variables x , nous avons intégré ces systèmes à l'aide de la fonction `ode45` de MATLAB[®]. Nous avons posé les conditions initiales $CI = [0, 0.1, 0.1]$, un pas d'intégration de 0.1 et un intervalle de temps $[0, 100]$ pour chacun des systèmes. Cela est motivé par le fait qu'à partir de ce pas, il n'y a plus d'amélioration notable dans l'intégration comme le montre la figure 2.9. Cela est surtout valable pour Lorenz. D'autres pas ont été essayés pour Rossler mais aucune "stagnation" n'a été trouvée et pour ne pas prendre trop de temps, nous choisissons également 0.1.

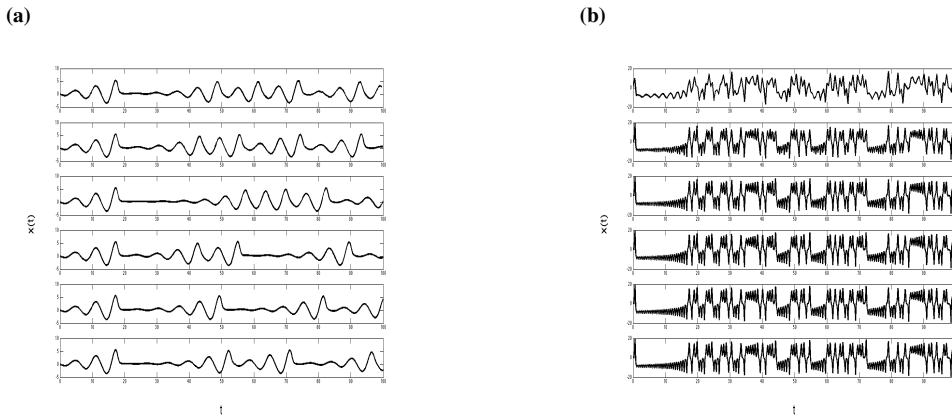


FIGURE 2.9 – (a) x —Rossler ; (b) x —Lorenz pour différents pas d'intégration sur l'intervalle de temps $[0, 100]$. De haut en bas : 0.5, 0.2, 0.1, 0.05, 0.02, 0.01. Pour (b) x —Lorenz, nous ne remarquons plus de différences notables dans les solutions à partir d'un pas 0.1.

En vue d'obtenir un nombre de quantiles optimal, nous regardons tout d'abord les distances entre les séries de première et seconde génération. Malheureusement, aucune tendance ne semble se dégager (figure 2.10). Les erreurs se valent pour les deux systèmes. Les valeurs de Rossler sont, grosso modo, de -4 à 6 et la distance moyenne est environ 2.7. Pour Lorenz, les valeurs oscillent entre -20 et 20 et distance moyenne est environ 11. C'est presque un tiers à chaque fois. Grâce aux bornes d'erreurs, nous remarquons que la distance peut varier assez fortement d'une réalisation à l'autre. Cela est dû au caractère aléatoire de la méthode.

Au niveau des réseaux, nous calculons d'autres quantités reprises à la figure 2.11. De façon générale, les valeurs obtenues pour Rossler sont plus élevées que celles de Lorenz et nous parvenons à mieux retrouver ces valeurs chez Lorenz à travers les différentes générations qu'avec Rossler. Plus particulièrement, quel que soit le nombre de quantiles, nous retrouvons à peu près à chaque fois la même modularité. Au niveau des self-loops, il y en a beaucoup plus chez Rossler et elles semblent être à chaque fois moins présentes dans le réseau de deuxième génération. Chez Lorenz, il y en a moins et cette quantité semble être plus constante au travers des générations. Le diamètre des réseaux augmente en fonction de \mathcal{Q} . Contrairement à Rossler, les réseaux de première et seconde générations de Lorenz possèdent à peu près le même diamètre.

Afin de déterminer un nombre optimal de quantiles, nous procédons à un classement sur les différences : pour chaque mesure, nous les classons par ordre croissant de leurs différences et leur score égale leur ranking. Pour Rossler, nous obtenons que $\mathcal{Q} = 10$ renvoie les meilleurs résultats tandis que pour Lorenz, $\mathcal{Q} = 50$ est le meilleur. Cependant, pour Rossler, le deuxième meilleur \mathcal{Q} est 50. De plus, l'écart entre $\mathcal{Q} = 10$ et $\mathcal{Q} = 50$ est beaucoup plus petit que ceux entre les autres \mathcal{Q} . Par conséquent, nous appliquons les cartes \mathcal{M}_{QT} et \mathcal{M}_{QT}^{-1} avec 50 quantiles.

Comme dit précédemment, le comportement de x -Rossler génère des oscillations quasi-périodiques qui se remarquent également dans la fonction d'autocorrélation ainsi que dans le spectre de puissance qui affiche une claire power-law scaling (figure 2.12). Malheureusement, nous n'arrivons pas à retrouver ce comportement lorsque nous reconstruisons le signal, même en faisant varier le nombre de quantiles. Cela s'explique par l'allure de la matrice d'adjacence du graphe de première génération. Nous voyons qu'il y a beaucoup trop de "choix" pour déterminer le quantile suivant. Bien que

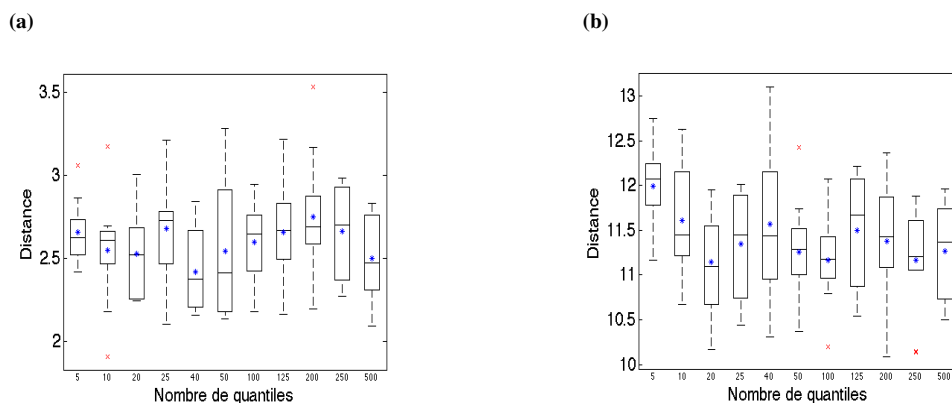


FIGURE 2.10 – Distances, en fonction du nombre de quantiles, entre les séries temporelles de première et seconde génération de (a) x -Rössler ; (b) x -Lorenz sur l'intervalle $[0, 100]$ en considérant un pas d'intégration de 0.1. Ces images sont obtenues sur base de 10 réalisations. La croix bleue représente la moyenne pour chaque \mathcal{Q} considéré. Aucune tendance ne semble se dégager. Les erreurs se valent pour les deux systèmes. Les valeurs de Rössler sont, *grosso modo*, de -4 à 6 et la distance moyenne est environ 2.7. Pour Lorenz, de -20 à 20 et distance 11. Donc c'est presque un tiers à chaque fois. Les bornes d'erreurs mettent en avant le fait que la distance peut varier assez fortement d'une réalisation à l'autre. Cela est dû au caractère aléatoire de la méthode.

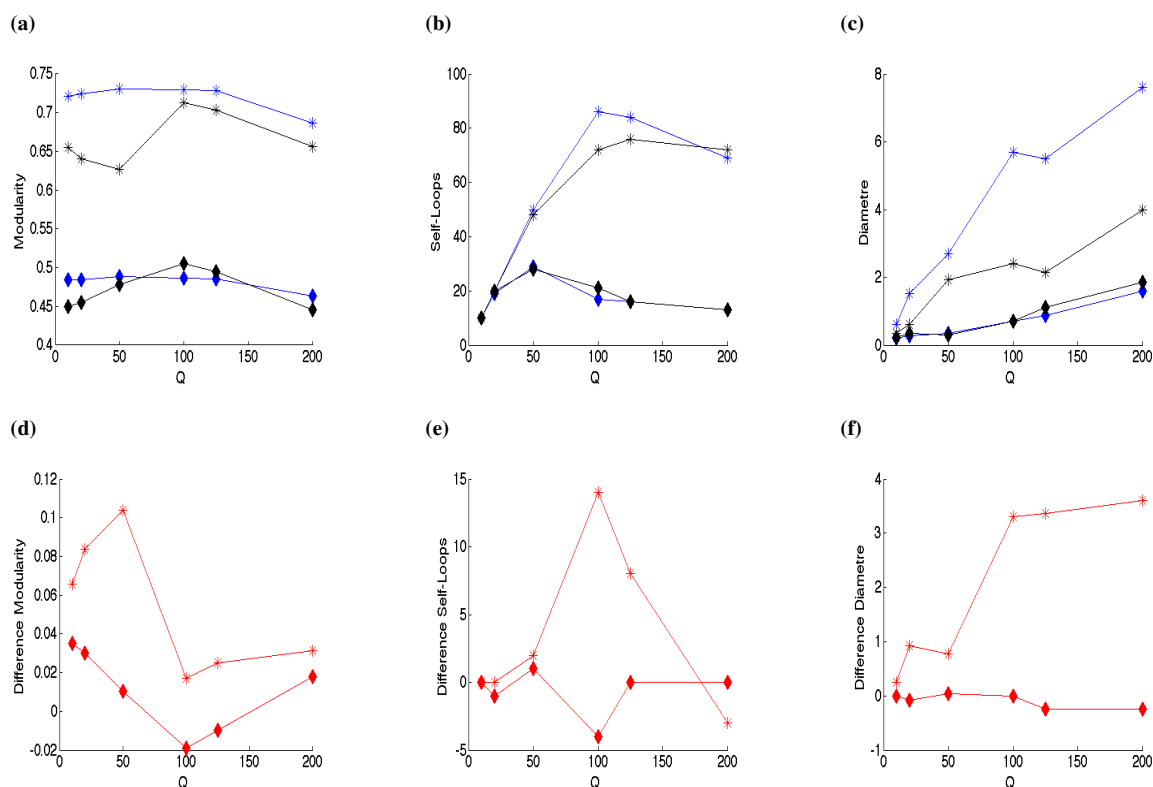


FIGURE 2.11 – (a) Modularité ; (b) Self-Loops ; (c) Diamètre des différents réseaux associés à x -Rössler et x -Lorenz obtenus en considérant différentes valeurs de \mathcal{Q} . Les croix représentent les valeurs obtenues pour Rössler tandis que les losanges sont reliés à Lorenz. En bleu les données pour la première génération, en noir pour la seconde. Les différences entre première et seconde générations des différentes mesures sont tracées sur les panels (d) à (f) respectivement. Seules sont représentées les moyennes, obtenues à partir de 10 réalisations. De façon générale, les valeurs obtenues pour Rössler sont plus élevées que celles de Lorenz et nous parvenons à mieux retrouver ces valeurs chez Lorenz à travers les différentes générations qu'avec Rössler. Plus particulièrement, quel que soit le nombre de quantiles, nous retrouvons à peu près à chaque fois la même modularité. Au niveau des self-loops, il y en a beaucoup plus chez Rössler et elles semblent être à chaque fois moins présentes dans le réseau de deuxième génération. Chez Lorenz, il y en a moins et cette quantité semble être plus constante au travers des générations. Le diamètre des réseaux augmente en fonction de \mathcal{Q} . Contrairement à Rössler, les réseaux de première et seconde générations de Lorenz possèdent à peu près le même diamètre.

les séries temporelles aient une allure assez différente, la répartition des valeurs des data est similaire. Quant à x -Lorenz, son allure est plus aléatoire. Les fonctions d'autocorrélation, les spectres de puissance (à noter que nous avons également une power-law, mais avec une pente plus douce) ainsi que la distribution des valeurs des signaux (figure 2.13) rendent compte de la similarité entre les deux générations.

Au niveau des graphes, le caractère quasi-périodique de la série temporelle de Rossler confère au graphe correspondant une allure de "chaîne". La modularité de ces graphes égale 0.7408 et 0.6962, ce qui est beaucoup plus grand que la modularité obtenue à partir de la randomisation de la série temporelle qui est de 0.1122 et 0.1483 respectivement. Par randomisation, nous entendons un réarrangement aléatoire des valeurs composant la série de base. À l'inverse, les graphes correspondant à Lorenz sont plus denses, ce qui est dû au caractère plus aléatoire de la série de base. Ces réseaux sont aussi modulaires, avec des modularités de 0.4889 et 0.4437, ce qui est également plus grand que celle obtenue après randomisation : 0.1127 et 0.1611. Les résultats ainsi obtenus montrent que des séries temporelles possédant des propriétés différentes produisent des réseaux ayant des topologies différentes. Les réseaux de Lorenz possèdent en réalité deux larges modules qui correspondent aux deux lobes de l'attracteur (figure 2.14). Pour la représentation des graphes, chaque nœud possède une taille proportionnelle à son degré. Chaque nœud est également coloré en fonction du module auquel il appartient. Afin de savoir quel quantile forme quel module, nous avons repris le même code couleur dans la représentation des signaux.

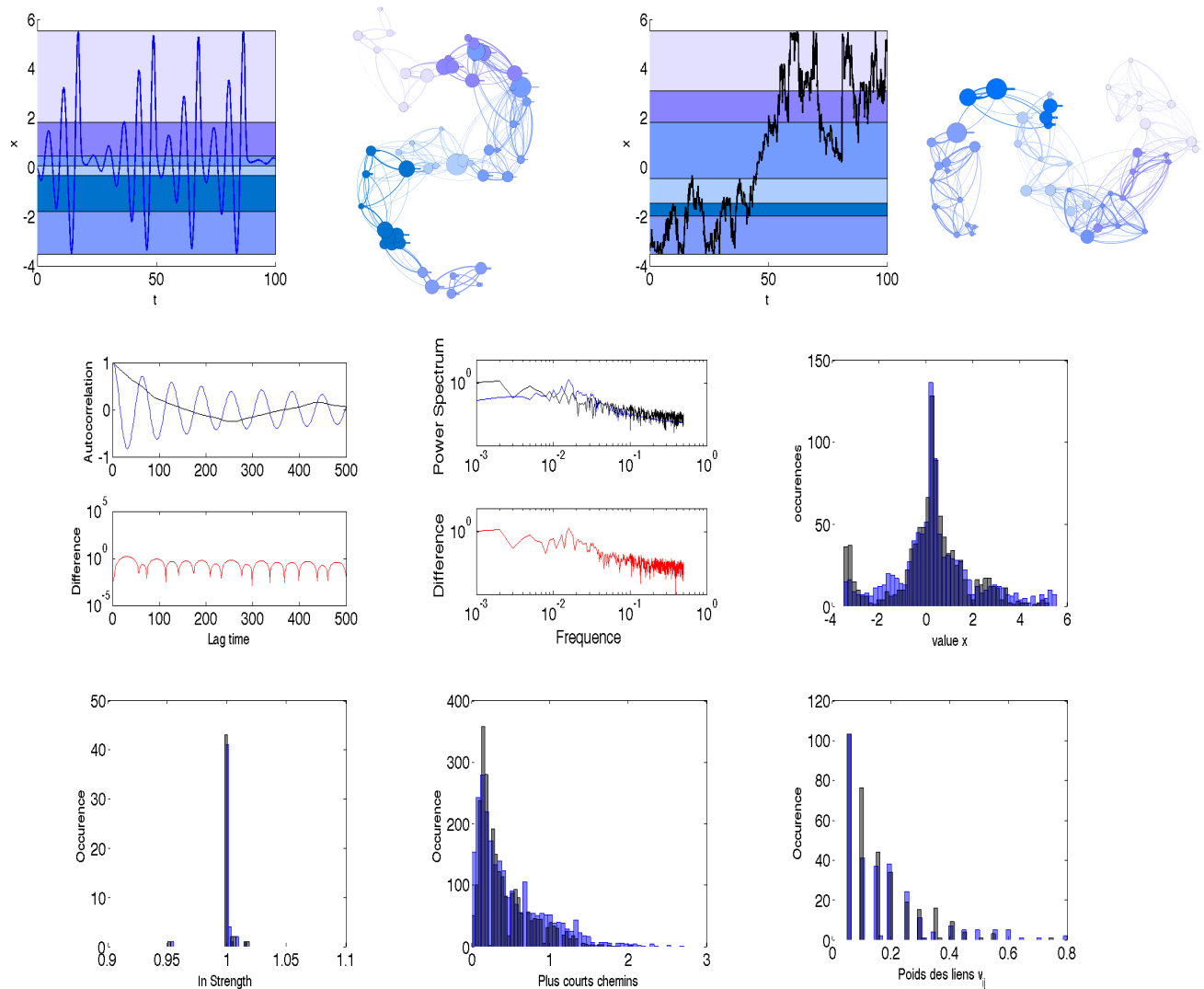


FIGURE 2.12 – Application des cartes \mathcal{M}_{QT} et \mathcal{M}_{QT}^{-1} sur x -Rossler accompagnée de quelques mesures de comparaison entre les séries temporelles et réseaux de première et seconde génération. Pour les histogrammes, les valeurs représentées sont les valeurs moyennes obtenues à partir de 10 réalisations. Le caractère quasi-périodique de la série temporelle de Rossler génère des oscillations quasi-périodiques qui se remarquent dans la fonction d'autocorrélation ainsi que dans le spectre de puissance qui affiche une claire power-law scaling et confère au graphe correspondant une allure de "chaîne". La modularité de ces graphes égale 0.7408 et 0.6962. Les valeurs envisagées pour comparer les graphes sont fort semblables à travers les différentes générations.

Division en intervalles de même taille

Nous appliquons la même philosophie : nous divisons le domaine de la série temporelle en différents intervalles et chaque intervalle correspond à un nœud dans le réseau associé, à la différence que maintenant, nous divisons le domaine en intervalles de même taille. Il est donc fort probable qu'il n'y ait plus le même nombre de points dans chacun d'eux. Pour reconstruire le signal, nous procédons toujours à une marche aléatoire sur le réseau. Les distances entre les signaux de première et deuxième générations sont reprises à la figure 2.15. Bien qu'il soit difficile de comparer de façon très précise avec la division en quantiles, nous remarquons une légère diminution de la distance moyenne. Au niveau des graphes, nous avons là quelques différences du fait que nous obtenons des nœuds moins connectés. Nous avons donc

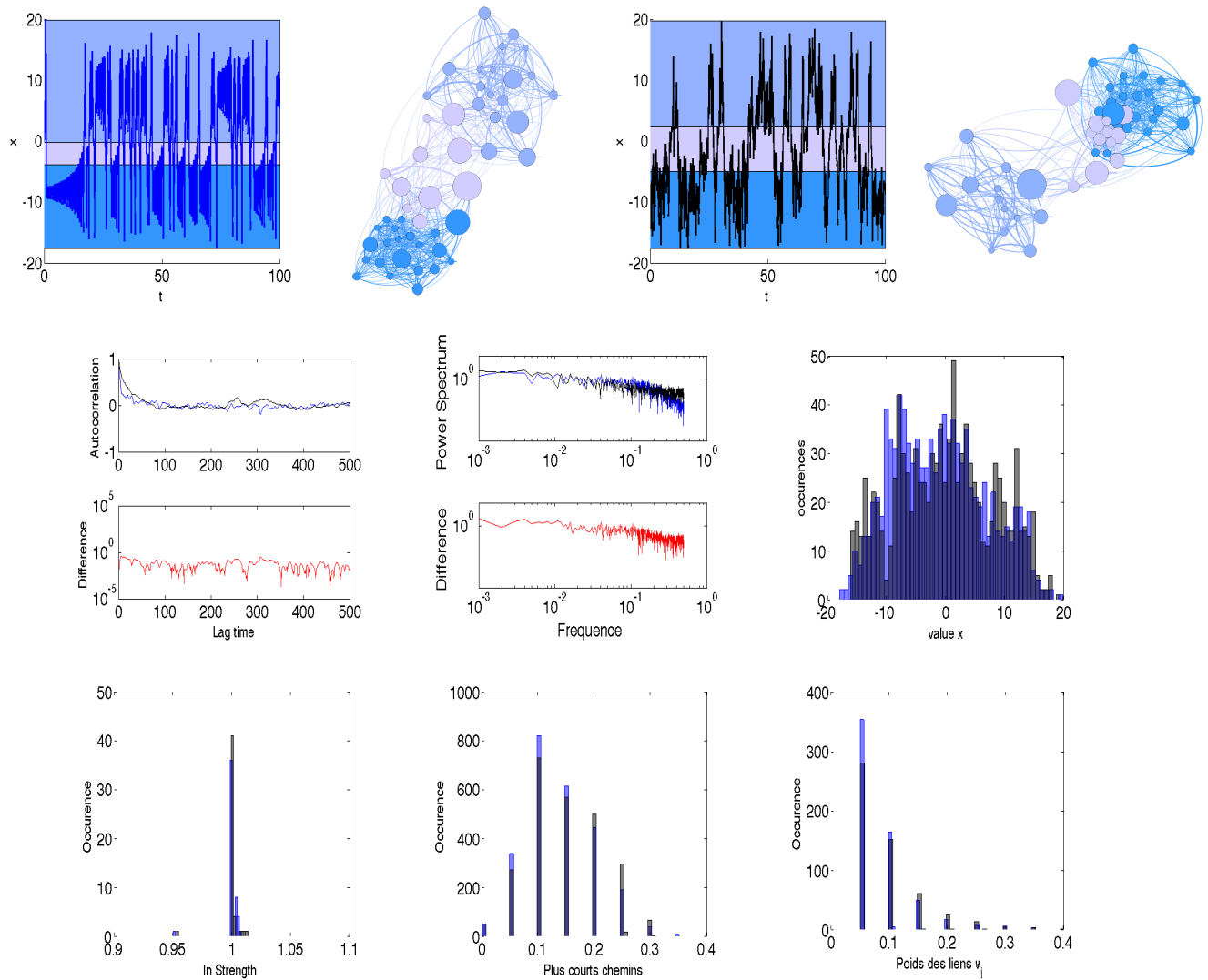


FIGURE 2.13 – Application des cartes M_{QT} et M_{QT}^{-1} sur x -Lorenz accompagnée de quelques mesures de comparaison entre les séries temporelles et réseaux de première et seconde génération. Pour les histogrammes, les valeurs représentées sont les valeurs moyennes obtenues à partir de 10 réalisations. Les réseaux obtenus sont relativement denses et possèdent une modularité de 0.4889 et 0.4437 pour la première et deuxième génération respectivement. Les valeurs envisagées pour comparer les graphes sont fort semblables à travers les différentes générations. Ces réseaux possèdent deux larges modules qui correspondent aux deux lobes de l'attracteur.

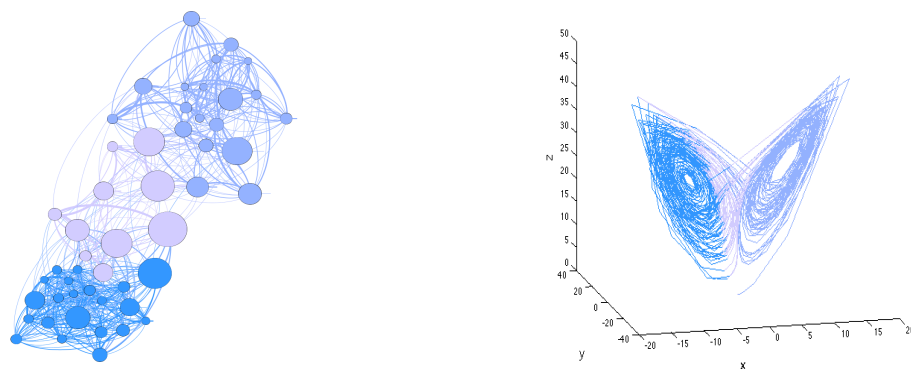


FIGURE 2.14 – Réseau associé à x -Lorenz et Attracteur de Lorenz. Les deux plus gros modules présents dans le réseau correspondent aux deux lobes de l'attracteur.

par exemple, des diamètres finis qui deviennent infinis (figure 2.16). Au vu de ces résultats, nous ne pouvons tirer un nombre optimal d'intervalles. Les figures 2.17 et 2.18 reprennent les résultats appliqués avec 50 intervalles. Nous remarquons, grâce aux mesures envisagées ici, que comparé aux quantiles, nous avons une petite détérioration au niveau des reconstructions.

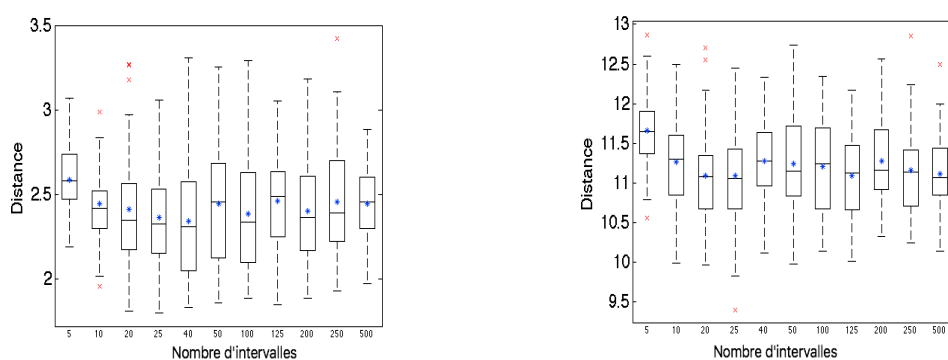


FIGURE 2.15 – Distances entre les séries de premières et secondes générations pour (a) Rossler ; (b) Lorenz calculées à partir de 10 réalisations. Le domaine des séries temporelles est divisé en N intervalles de même longueur. Ces distances sont légèrement plus faibles que pour la division du domaine de la série en quantiles (figure 2.10).

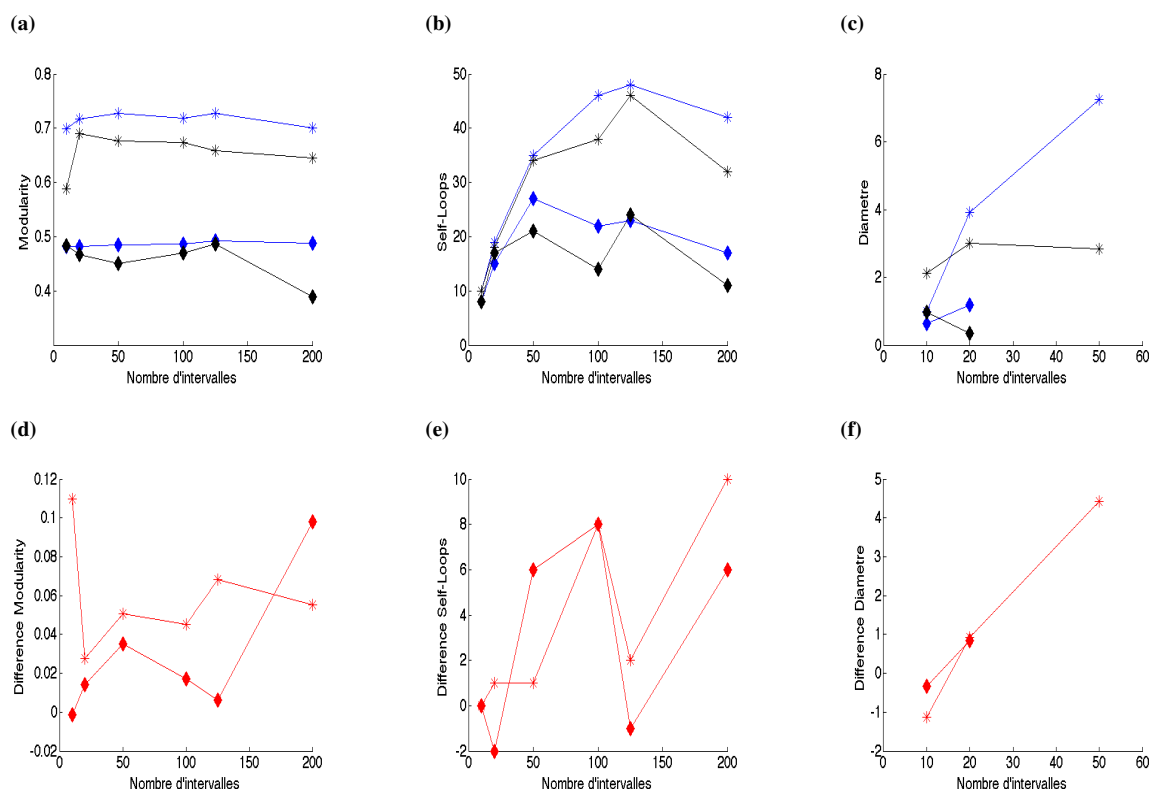


FIGURE 2.16 – (a) Modularité ; (b) Self-Loops ; (c) Diamètre des différents réseaux obtenus en considérant différentes valeurs pour le nombre d'intervalles pris dans le domaine de la série temporelle. Les croix représentent les valeurs obtenues pour Rossler tandis que les losanges sont reliés à Lorenz. En bleu les données pour la première génération, en noir pour la seconde. Les différences entre première et seconde générations des différentes mesures sont tracées sur les panels (d) à (f) respectivement. Seules sont représentées les moyennes, obtenues à partir de 10 réalisations. De façon générale, les valeurs obtenues pour Rossler sont plus élevées que celles de Lorenz. Au niveau des self-loops, il y en a beaucoup plus chez Rossler et elles semblent être à chaque fois moins présentes dans le réseau de deuxième génération. Le diamètre des réseaux augmente en fonction du nombre d'intervalles. Il arrive cependant que le diamètre ne soit pas défini (infini) ; pour Rossler : au-delà de $\mathcal{Q} = 50$ tandis que pour Lorenz : au-delà de $\mathcal{Q} = 20$.

Changement du pas d'intégration

Nous intégrons à présent les différents systèmes avec différents pas d'intégration : 0.05, 0.02 et 0.01. Ces valeurs sont toutes inférieures à 0.1, ce qui nous donnera une "bonne" intégration (figure 2.9). Comme précédemment, nous calculons les distances entre les séries générées. Les figures 2.19 et 2.20 représentent la distance euclidienne entre les séries temporelles de première et seconde génération pour x -Rossler et x -Lorenz respectivement. Les figures sont obtenues sur base de 10 réalisations. Pour ne pas devoir calculer de déphasage, nous avons effectué la marche aléatoire à partir du nœud du premier point de la série de base. Changer le pas d'intégration n'intervient pas vraiment dans la distance. Pour Rossler, lorsque le nombre de quantiles change, il semble avoir une petite tendance : "plus de quantiles, plus grande distance". Pour Lorenz, les distances sont relativement stables. Nous observons que la distance maximale moyenne est toujours pour 5 quantiles. Mais de façon générale, le nombre de quantile ne semble pas intervenir de façon cruciale dans

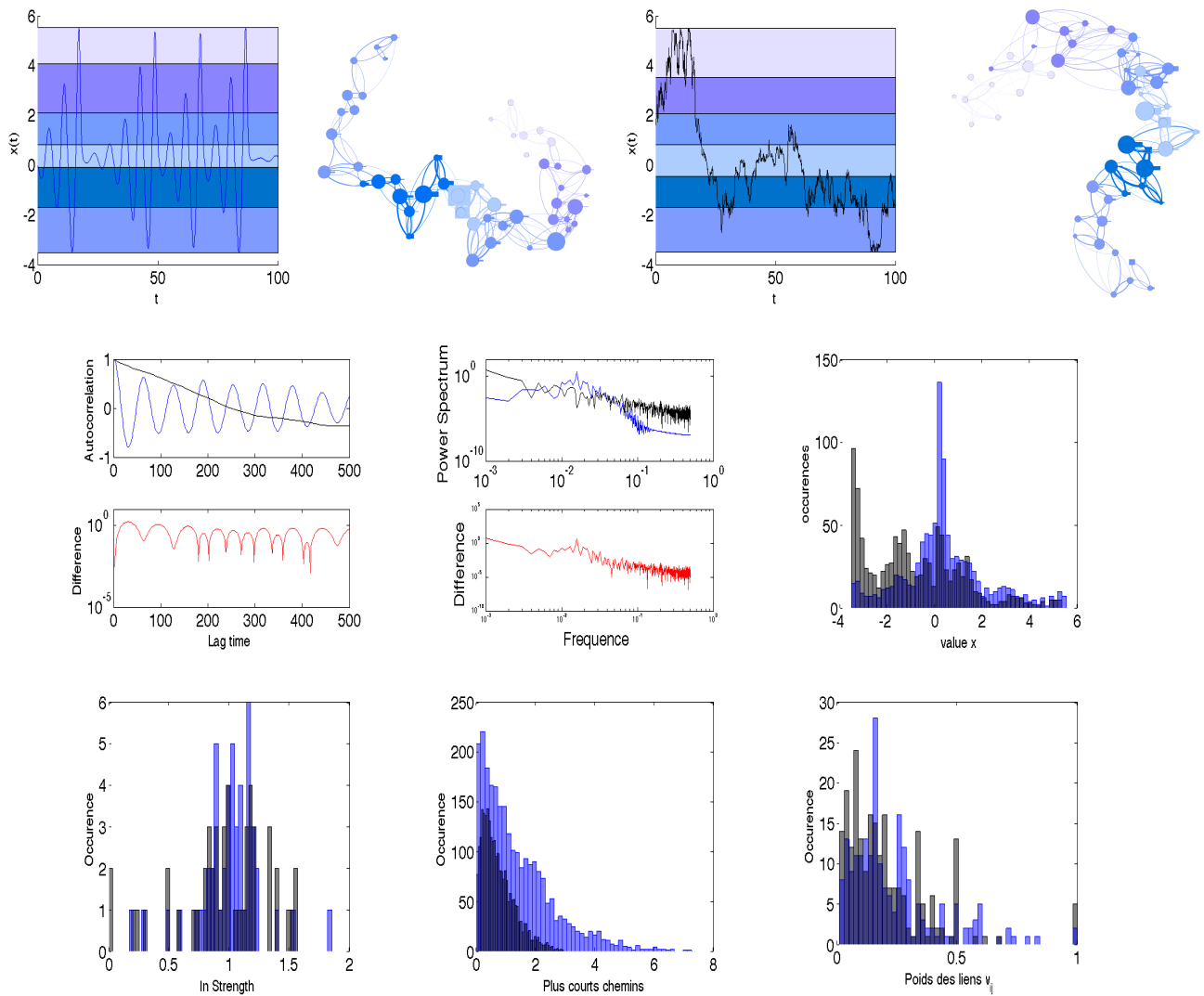


FIGURE 2.17 – Application des cartes \mathcal{M}_{QT} et \mathcal{M}_{QT}^{-1} sur x -Rossler accompagnée de quelques mesures de comparaison entre les séries temporelles et réseaux de première et seconde génération. Le domaine des séries temporelles est divisé en 50 intervalles de même longueur. Pour les histogrammes, les valeurs représentées sont les valeurs moyennes obtenues à partir de 10 réalisations. Comparé aux quantiles, nous avons une petite détérioration au niveau des reconstructions des séries et graphes de seconde génération ; cela se remarque surtout dans les comparaisons des mesures de graphes (plus courts chemins). Les graphes possèdent toujours une structure en chaîne.

le processus de reconstruction.

Une diminution dans le pas d'intégration se remarque fort dans la construction du réseau et donc dans la reconstruction du signal. En effet, le fait d'intégrer de façon plus précise implique que les signaux sont composés de plus de points (pour un même intervalle de temps). Dès lors, la différence entre un point et son successeur est réduite. Pour un nombre de quantiles fixé, cela signifie que la transition d'un point au suivant a plus de chance de se faire au sein d'un même quantile, ou du moins dans un quantile voisin. Cela se traduit par une matrice d'adjacence où les entrées se concentrent plutôt sur la diagonale et les sub/sup diagonales. Cela est illustré à la figure 2.21 sur x -Lorenz avec $\mathcal{Q} = 50$.

Le réseau est donc plus "en chaîne" quel que soit le nombre de quantiles considéré. Il ne faut donc pas spécialement prendre plus de quantiles lorsque le pas est plus petit puisque la reconstruction de la série n'est pas impactée par ce choix. Nous appliquons encore une fois les cartes \mathcal{M}_{QT} et \mathcal{M}_{QT}^{-1} sur nos différents systèmes avec 50 quantiles mais cette fois, nous prenons un pas d'intégration de 0.01 (figure 2.22). Nous remarquons que, quelle que soit la série temporelle de base, nous obtenons un réseau en chaîne. Cette fois, nous ne savons plus vraiment faire de différences entre les résultats générés par x -Rossler et x -Lorenz. La figure 2.23 reprend les différentes matrices d'adjacence obtenues en considérant différents \mathcal{Q} . Lorsque le nombre de quantiles est bas, i.e. égale à 20, les réseaux résultants de Rossler et Lorenz possèdent quasi la même matrice d'adjacence. En augmentant le nombre de quantiles, une légère différence se remarque même si l'allure des réseaux sera toujours en chaîne. Il semblerait donc que ce qui importe le plus (pour obtenir des réseaux "différents" à partir de signaux "différents") soit le nombre de points pris par intervalle de temps.

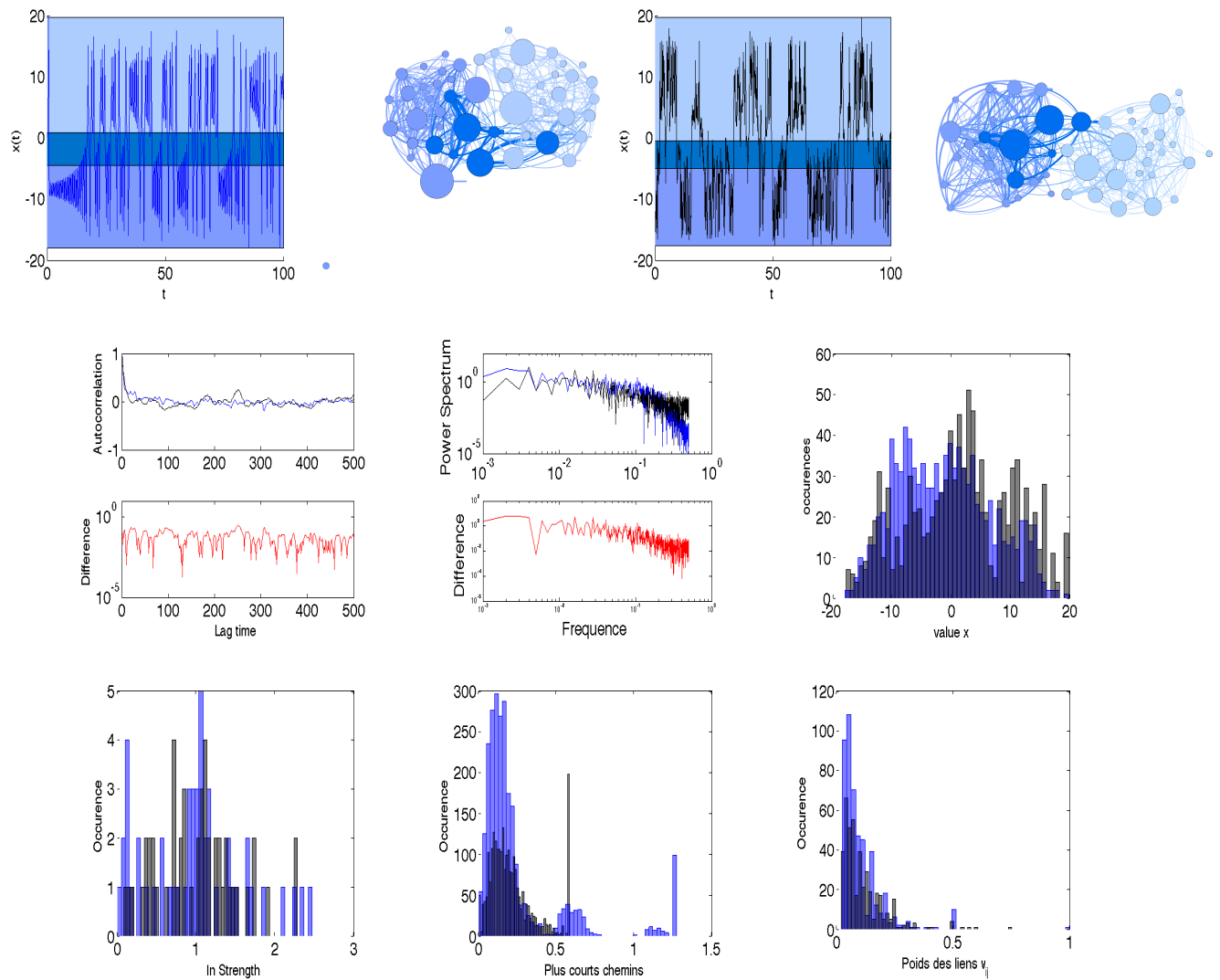


FIGURE 2.18 – Application de \mathcal{M}_{QT} et \mathcal{M}_{QT}^{-1} sur x -Lorenz accompagnée de quelques mesures de comparaison entre les séries temporelles et réseaux de première et seconde génération. Le domaine des séries temporelles est divisé en 50 intervalles de même longueur. Pour les histogrammes, les valeurs représentées sont les valeurs moyennes obtenues à partir de 10 réalisations. Comparé aux quantiles, nous avons une petite détérioration au niveau des reconstructions des séries et graphes de seconde génération ; cela se remarque surtout dans les comparaisons des mesures de graphes (plus courts chemins). Les graphes résultants sont relativement denses.

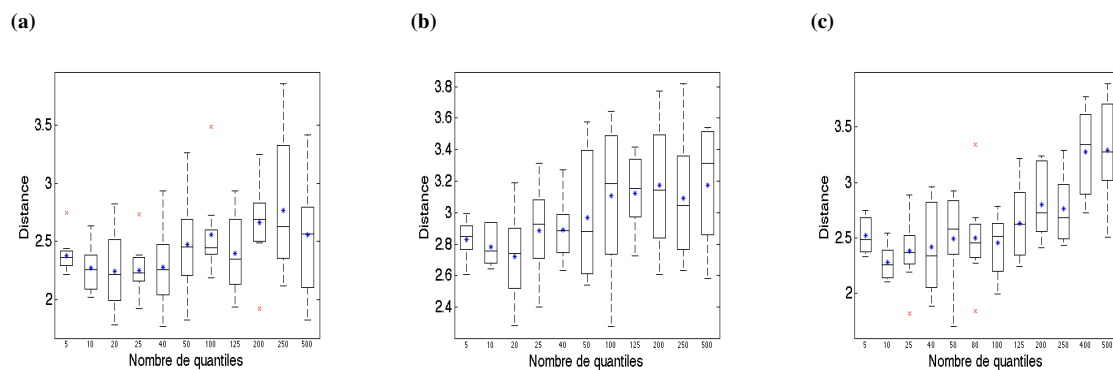


FIGURE 2.19 – Distances entre les séries temporelles de première et seconde génération de x -Rosler sur l'intervalle $[0, 100]$ en considérant un pas d'intégration de (a) 0.05 ; (b) 0.02 et (c) 0.01. Les croix bleues représentent les moyennes. Il semble avoir une petite tendance, relativement faible : "plus de quantiles, plus grande distance"

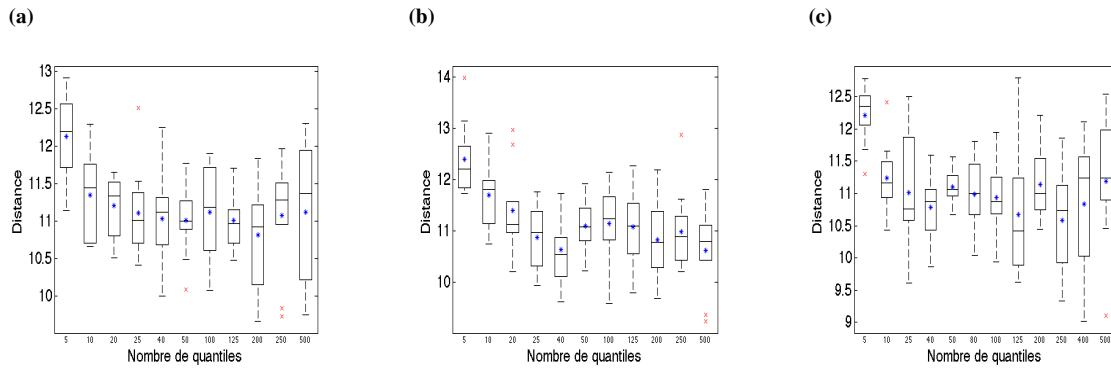


FIGURE 2.20 – Distances entre les séries temporelles de première et seconde génération de x -Lorenz sur l'intervalle $[0, 100]$ en considérant un pas d'intégration de (a) 0.05 ; (b) 0.02 et (c) 0.01. Les croix bleues représentent les moyennes.

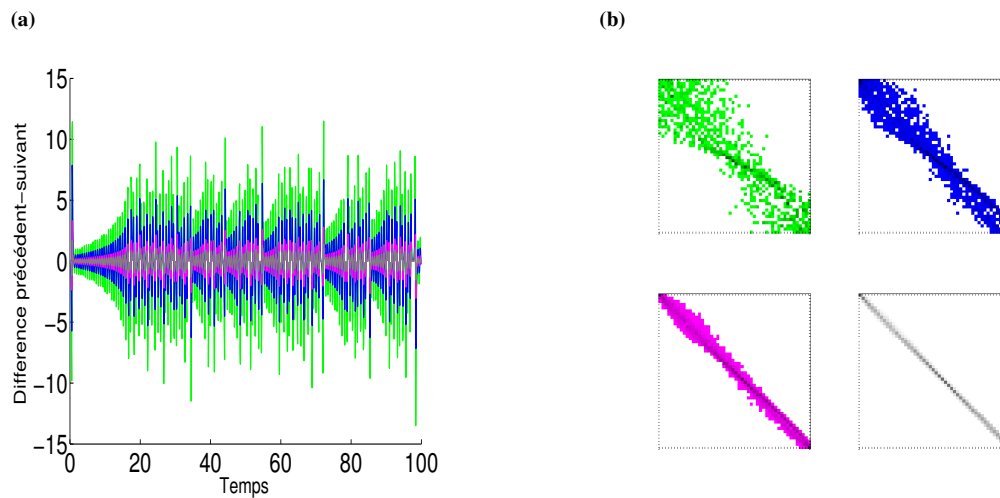


FIGURE 2.21 – (a) Différence entre deux valeurs successives dans ; (b) Matrices d'adjacence associées à x -Lorenz pour différents pas d'intégration sur l'intervalle $[0, 100]$ en considérant 50 quantiles. Vert : 0.1, bleu : 0.05, magenta : 0.02, gris : 0.01. Une diminution dans le pas d'intégration implique que la différence entre un point et son successeur est réduite. Cela se traduit par une matrice d'adjacence où les entrées se concentrent plutôt sur la diagonale et les sub/sup diagonales.

2.2.3 Chaîne de Markov d'ordre supérieur

De façon générale, bien que quelques mesures soient similaires entre les générations, nous ne parvenons pas à trouver un "nec plus ultra" comme avec la série périodique (eq 2.2), que ce soit avec les quantiles ou avec des intervalles de même longueur. Cela vient du fait que la méthode est fort aléatoire et lors de la reconstruction, nous avons beaucoup trop de choix. Une façon de réduire ce choix et par là-même d'améliorer la justesse de la reconstruction est de considérer une chaîne de Markov d'ordre supérieur à 1. Jusqu'à présent, nous n'avons considéré que ce type de chaîne dont une définition nous est donnée par [44] : La suite (X_1, X_2, \dots, X_n) à valeurs dans un ensemble E est une chaîne de Markov d'ordre 1 si pour tout $1 < k \leq n$, $x_1, x_2, \dots, x_k \in E$,

$$P(X_k = x_k | X_{k-1} = x_{k-1}, \dots, X_1 = x_1) = P(X_k = x_k | X_{k-1} = x_{k-1}). \quad (2.8)$$

Dans cette partie, nous envisageons de prendre une chaîne d'ordre supérieur, ce qui s'énonce en généralisant (2.8) : La suite (X_1, X_2, \dots, X_n) à valeurs dans un ensemble E est une chaîne de Markov d'ordre l ($l \geq 1$) si pour tout $k > l$, $x_1, x_2, \dots, x_k \in E$,

$$P(X_k = x_k | X_{k-1} = x_{k-1}, \dots, X_1 = x_1) = P(X_k = x_k | X_{k-1} = x_{k-1}, \dots, X_{k-l} = x_{k-l}). \quad (2.9)$$

Dès lors, un état ne dépend plus uniquement de l'état précédent, mais des l états précédents. Remarquons qu'une chaîne de Markov d'ordre l sur E définie par les probabilités de transitions $P(X|Y)$ avec $X \in E$ et $Y \in E^l$ peut être ramenée à une chaîne de Markov d'ordre 1 sur E^l . La matrice de transition de cette dernière est $Q = (q_{YZ})$ avec $Y = (Y_1, \dots, Y_l) \in E^l$ et $Z = (Z_1, \dots, Z_l) \in E^l$ et est donnée par

$$q_{YZ} = q(Z|Y) := \delta_{Z_1 Y_2} \delta_{Z_2 Y_3} \dots \delta_{Z_{l-1} Y_l} P(Z_l | Y) \quad (2.10)$$

avec δ le symbole de Kronecker.

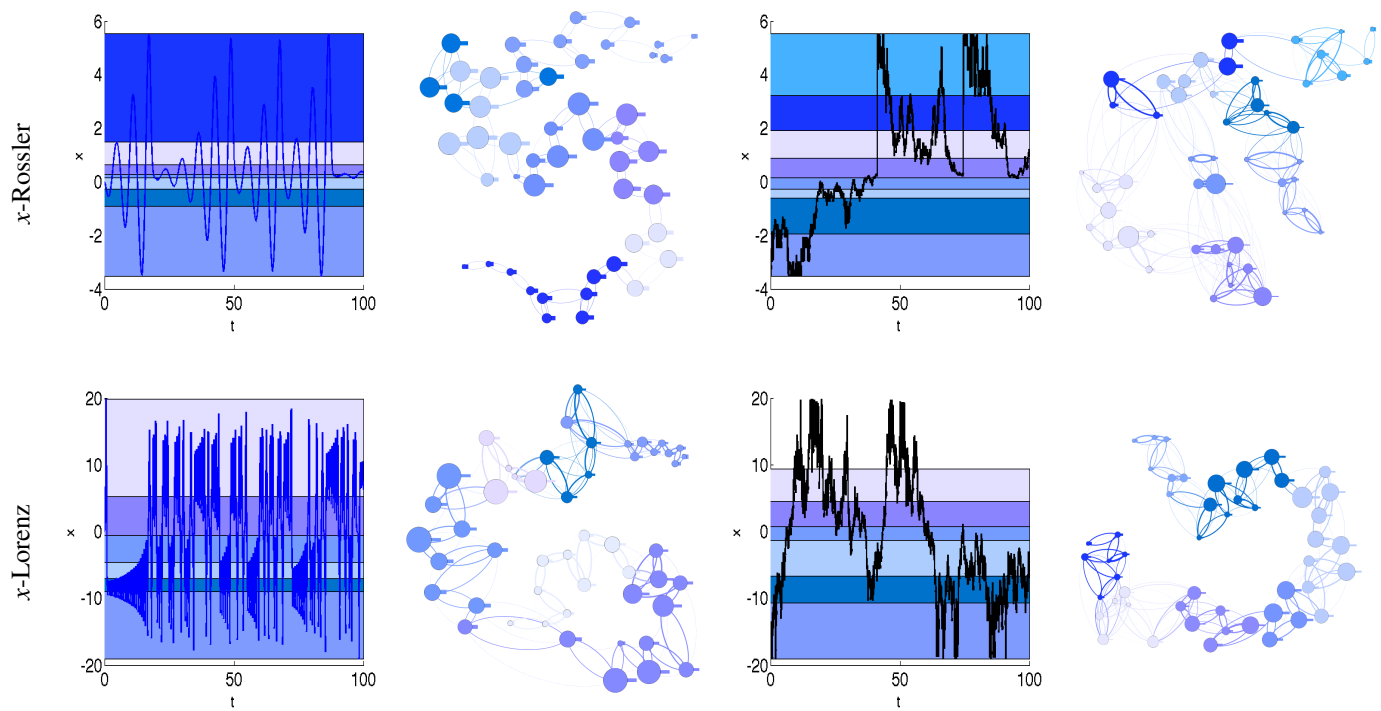


FIGURE 2.22 – Application des cartes \mathcal{M}_{QT} et \mathcal{M}_{QT}^{-1} sur x -Rossler et x -Lorenz avec un pas d'intégration 0.01 et 50 quantiles. Quelle que soit la série temporelle de base, le réseau obtenu possède une structure de "chaîne"; nous ne savons plus vraiment faire de différences entre les résultats générés par x -Rossler et x -Lorenz.

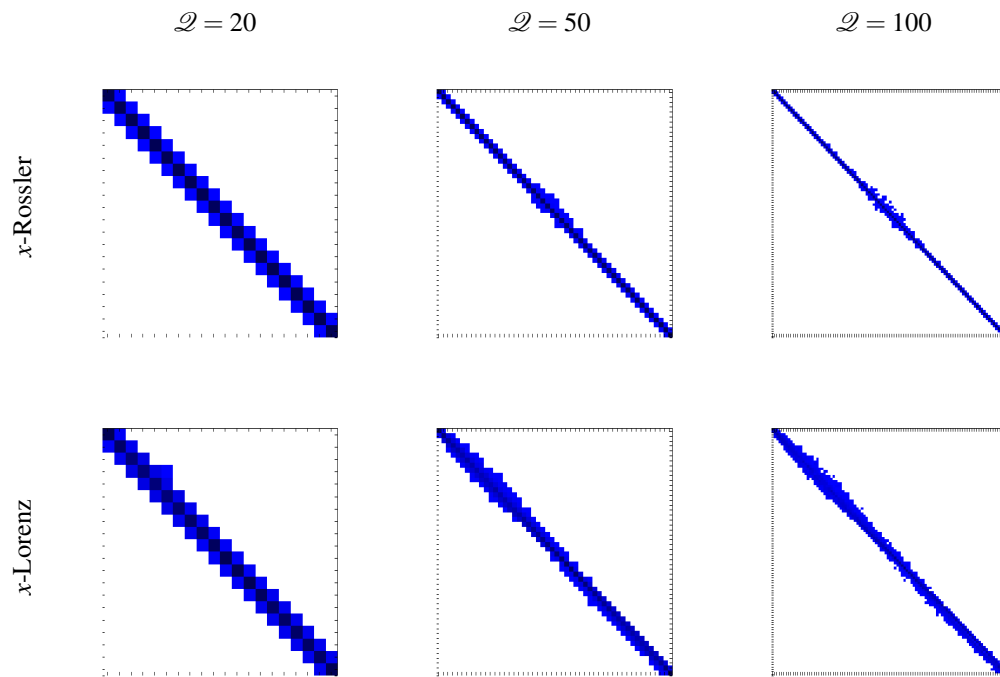


FIGURE 2.23 – Matrices d'adjacences des graphes de premières générations de Rossler et Lorenz pour différents nombres de quantiles : 20, 50 et 100 obtenues lorsque les systèmes sont intégrés avec un pas de 0.01.

Il existe plusieurs façons de représenter les graphes. Nous pouvons, comme mentionné ci-dessus, transformer la chaîne de Markov d'ordre 2 en une d'ordre 1 i.e. $((X_1, X_2), (X_3, X_4), \dots, (X_{n-1}, X_n))$ est une chaîne de Markov d'ordre 1 et ainsi considérer chaque couple de quantiles comme un nœud. Cependant, nous avons gardé ici la représentation un nœud pour un quantile. Deux nœuds i et j sont reliés non seulement s'il existe un point dans le quantile q_i dont son successeur est dans q_j mais également lorsque le successeur du successeur d'un point de q_i est dans q_j . Nous obtenons donc des réseaux "hyper-reliés". Cependant, la matrice des poids utilisée pour l'application \mathcal{M}_{QT}^{-1} est celle de dimension $\mathcal{Q}^2 \times \mathcal{Q}$ et non la matrice d'adjacence du réseau représenté.

Nous avons appliqué cela aux systèmes de Rossler et de Lorenz. En considérant un ordre 2, nous avons déjà une grande amélioration comme le montrent les figures 2.24 et 2.25. Nous pouvons bien entendu monter dans les ordres mais cela ne vaut pas la peine, en regard du gain par rapport au temps que cela prend. En moyenne, pour obtenir séries et graphes de première et seconde génération avec 50 quantiles, cela prend 0.9423 seconde pour l'ordre 1, 1.7880 seconde pour l'ordre 2 et déjà 802.3560 secondes pour l'ordre 3. Cela n'est pas étonnant puisque le nombre d'éléments à considérer croît de façon exponentielle. Dans la suite, nous nous contenterons donc de l'ordre 2. En particulier, nous définissons une matrice de transition $\mathcal{Q}^2 \times \mathcal{Q}$ qui donne la probabilité de X_{k+1} sachant le couple (X_{k-1}, X_k) .

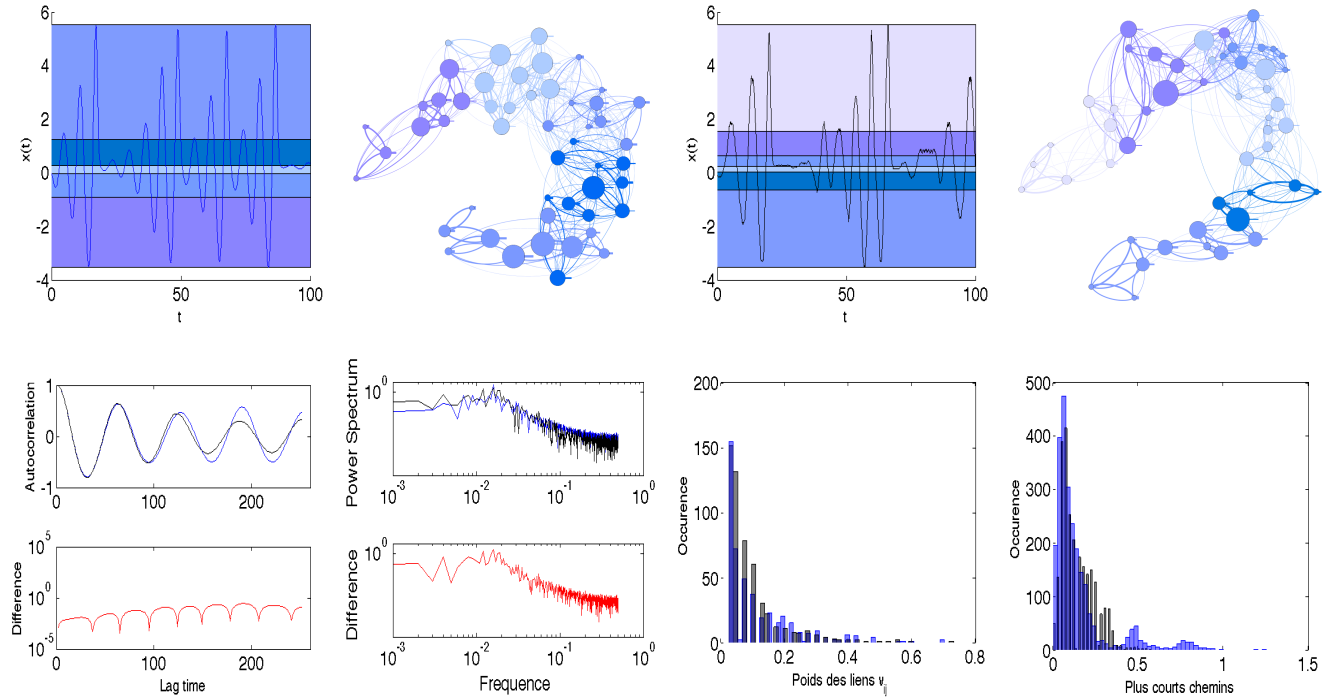


FIGURE 2.24 – Application de \mathcal{M}_{QT} et \mathcal{M}_{QT}^{-1} sur x -Rossler en considérant $Q = 50$ et une chaîne de Markov d'ordre 2 accompagnée de quelques comparaisons entre les signaux et réseaux. Les réseaux obtenus sont à présents hyper connectés et sont en forme de "chaîne" et sont modulaires ; leur modularité vaut 0.6802 et 0.6794 pour la première et seconde génération respectivement. Pour les histogrammes, les valeurs représentées sont les valeurs moyennes obtenues à partir de 10 réalisations. Les fonctions d'autocorrélation ainsi que les spectres de puissance sont beaucoup plus semblables que pour les chaînes de Markov d'ordre 1. Les mesures des graphes vont également dans le même sens.

À nouveau, nous regardons un peu le rôle des quantiles. La figure 2.26 reprend les distances entre les premières et secondes générations de signaux sur base de 50 réalisations. Tout d'abord, fait surprenant, nous ne remarquons pas de diminution de valeurs des distances par rapport à avant (chaîne de Markov d'ordre 1) mais surtout, nous remarquons que, pour Lorenz, les distances augmentent avec le nombre de quantiles. Cela peut s'expliquer par le fait que, lorsque \mathcal{Q} est plus grand, nous avons moins de liberté dans la reconstruction du signal. Différentes réalisations en fonction du nombre de quantiles (5, 20, 50 et 100) sont reprises à la figure 2.27. Bien que les distances augmentent (ou du moins, restent constantes), nous remarquons une amélioration dans la reconstruction du signal au plus le nombre de quantiles augmente. En réalité, au plus \mathcal{Q} est élevé, au plus nous avons de chance d'avoir des comportements comme entourés en rouge sur la figure 2.27. Dans le signal de base, nous retrouvons ces comportements mais ci ces derniers ne figurent pas exactement au même endroit dans les secondes générations que dans les premières, cela engendre un décalage et par conséquent, même si l'allure des signaux est beaucoup plus semblable, les distances augmentent. En résumé, lorsque le nombre de quantiles \mathcal{Q} est petit, considérer un ordre supérieur n'a pas beaucoup d'intérêt car cela reste encore trop aléatoire. Mais au plus \mathcal{Q} est grand, au plus cela permet de retrouver des "motifs" de la série temporelle de base et par là-même, d'améliorer la reconstruction. Malheureusement, nous venons de voir que ces "motifs" ne se reproduisent pas exactement au même moment que dans la série de base et donc, prendre la distance pour comparer les signaux n'est certainement pas le choix le plus judicieux.

Au niveau des graphes, nous comparons les différentes générations au travers de quelques mesures. La figure 2.28 reprend ces différentes analyses. Dans un premier temps, nous remarquons des plus petites différences entre les générations que lorsque l'ordre 1 était considéré. Mais ce qui importe le plus c'est qu'au plus le nombre de quantiles augmente, au plus les différences s'amenuisent sauf pour les boucles. Les valeurs sont exactement identiques pour 5, 10, 20, 50 quantiles et varie très légèrement pour 100 (au plus, une différence de 2 sur 80). Le nombre de quantiles joue un rôle crucial dans la méthode lorsque nous considérons des chaînes de Markov d'ordre supérieur. Il faut donc trouver un bon compromis entre la qualité des reconstructions et le temps que cela prend. La figure 2.29 reprend les temps d'exécution de la méthode pour obtenir les signaux et réseaux de premières et secondes générations en fonction du nombre de

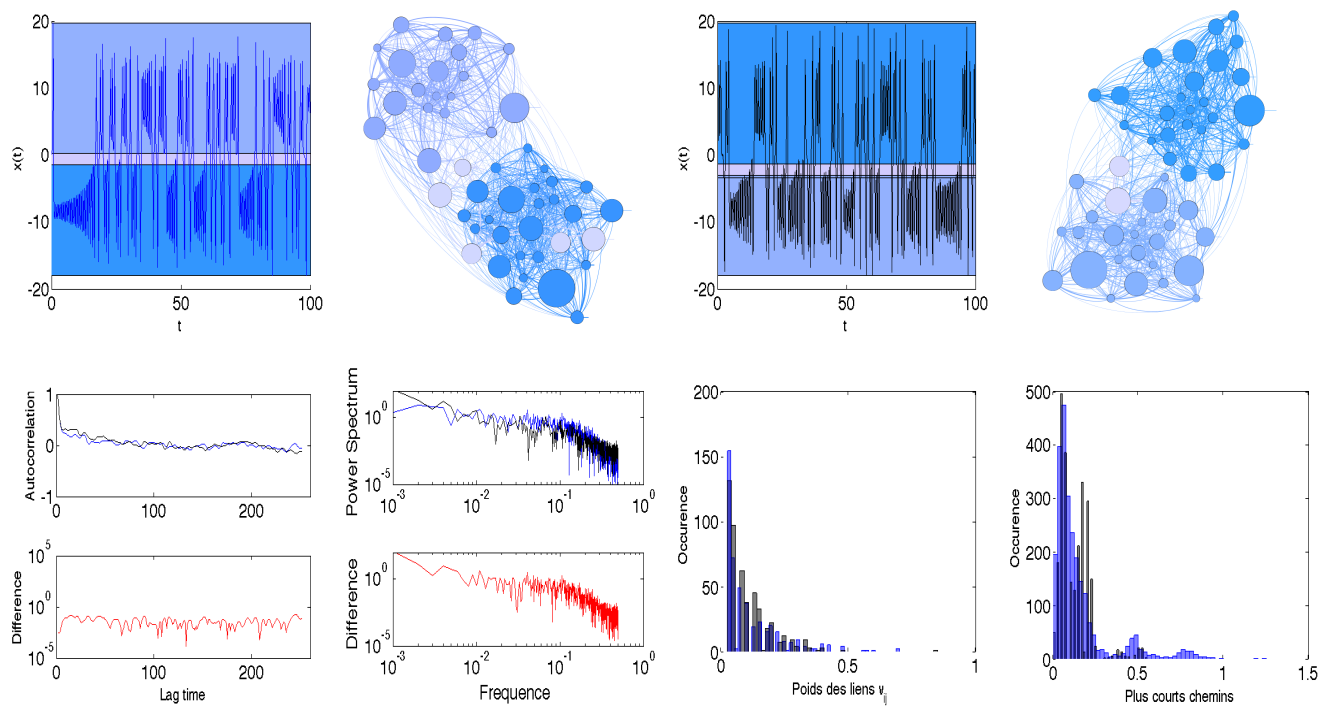


FIGURE 2.25 – Application de \mathcal{M}_{QT} et \mathcal{M}_{QT}^{-1} sur x -Lorenz en considérant $Q = 50$ et une chaîne de Markov d'ordre 2 accompagnée de quelques comparaisons entre les signaux et réseaux. Les réseaux obtenus sont à présents hyper connectés et sont relativement denses et sont modulaires ; leur modularité vaut 0.4298 et 0.4201 pour la première et seconde génération respectivement. Pour les histogrammes, les valeurs représentées sont les valeurs moyennes obtenues à partir de 10 réalisations. Les fonctions d'autocorrélation ainsi que les spectres de puissance sont beaucoup plus semblables que pour les chaînes de Markov d'ordre 1. Les mesures des graphes vont également dans le même sens.

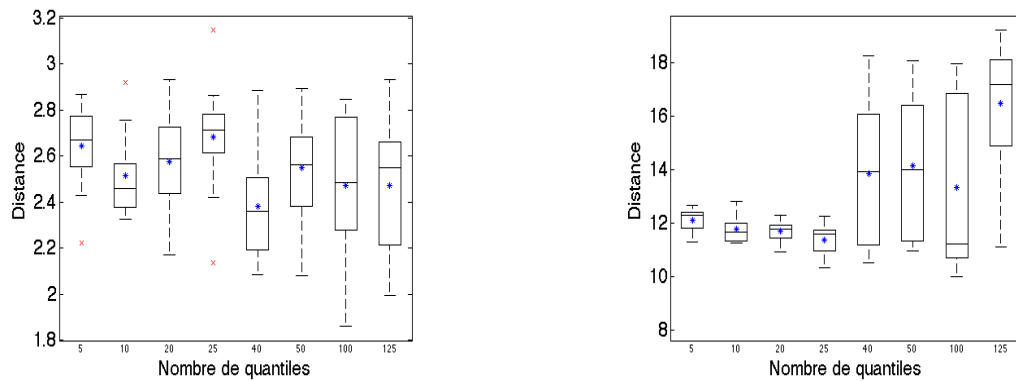


FIGURE 2.26 – Distances, en fonction du nombre de quantiles, entre les séries de premières et secondes générations en considérant une chaîne de Markov d'ordre 2. À gauche pour x -Rosler et à droite pour x -Lorenz. Ces résultats sont obtenus à partir de 50 réalisations. Les croix bleues dénotent les moyennes. Pour Rosler, les distances restent à peu près constantes quel que soit le nombre de quantiles. À l'inverse, pour Lorenz, il semble y avoir une légère diminution (et une faible variation) jusque 25 quantiles mais cette distance (et variation) augmente lorsque plus de quantiles sont considérés.

quantiles. Ici, nous ne considérons que de petits réseaux donc, même pour l'ordre 2 et 200 quantiles le temps ne s'élève qu'à 40 secondes mais pour des réseaux plus importants, le temps doit être pris en compte. Bien que les graphes soient maintenant hyper connectés, nous remarquons toujours que celui associé à Rosler est plus en chaîne que celui associé à Lorenz ; augmenter l'ordre de la chaîne de Markov ne semble pas altérer cette "propriété" déjà observée auparavant.

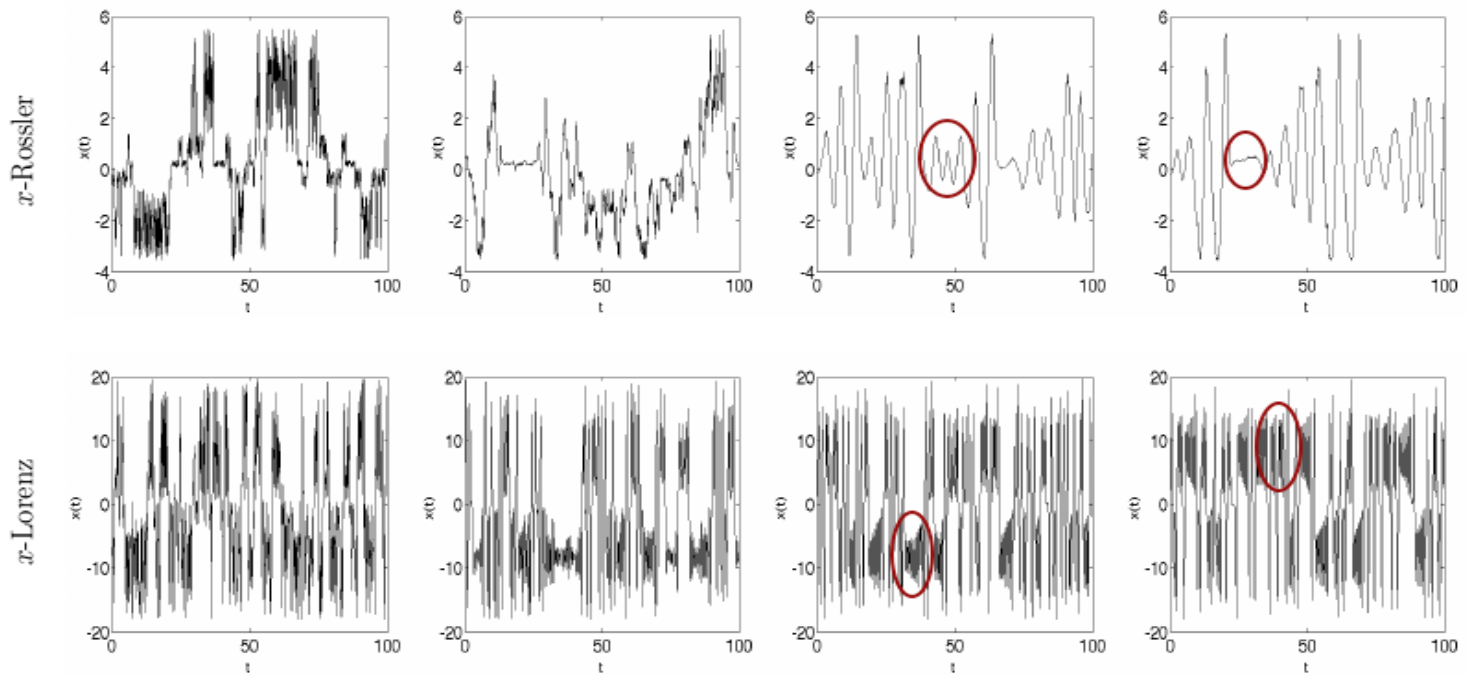


FIGURE 2.27 – Séries temporelles x -Rosler et x -Lorenz de secondes générations pour différents nombre de quantiles. Ces séries sont obtenues en considérant des chaînes de Markov d'ordre 2. De gauche à droite : 5, 20, 50 et 100 quantiles. Les cercles rouges entourent les "motifs problématiques" : des motifs qui sont également présents dans la série temporelle de base mais qui ne se situent pas exactement au même moment. C'est en partie à cause de ce décalage que les distances entre les signaux croient ou du moins, restent stables (figure 2.26) alors que le signal est de mieux en mieux reconstruit.

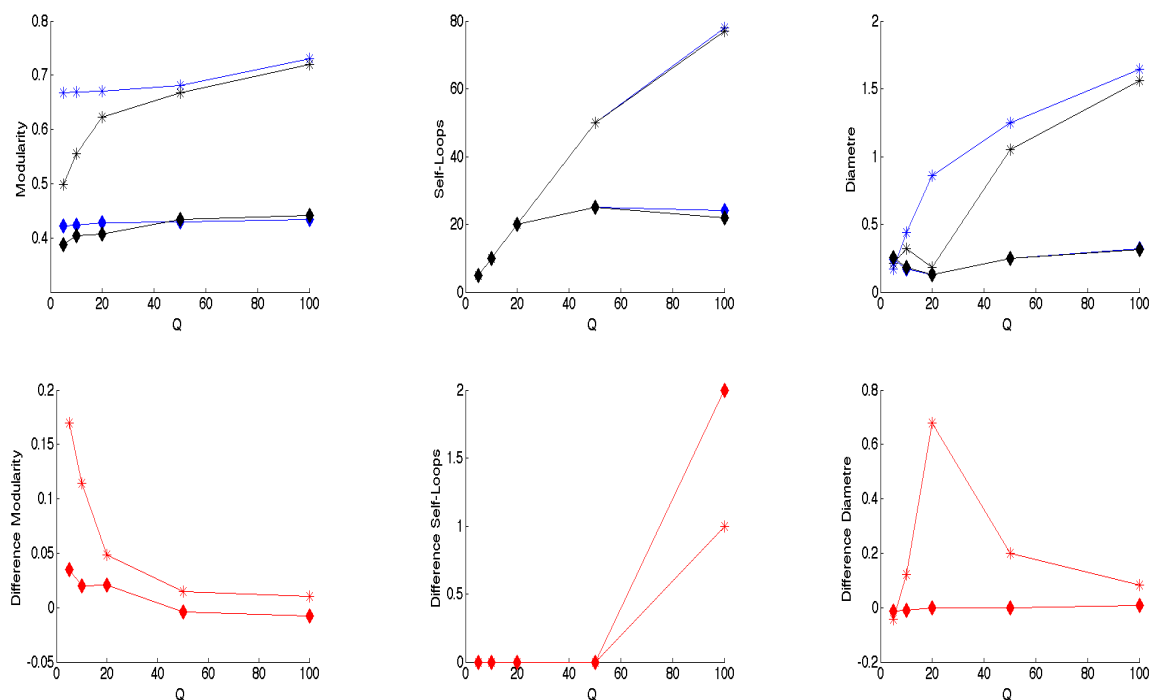


FIGURE 2.28 – (a) Modularité ; (b) Self-Loops ; (c) Diamètre des différents réseaux obtenus en considérant différentes valeurs de Q et une chaîne de Markov d'ordre 2. Les croix représentent les valeurs obtenues pour Rossler tandis que les losanges sont reliés à Lorenz. En bleu les données pour la première génération, en noir pour la seconde. Les différences entre première et seconde générations des différentes mesures sont tracées sur les panels (d) à (f) respectivement. Seules sont représentées les moyennes, obtenues à partir de 10 réalisations. Les différences sont beaucoup plus ténues que pour des chaînes de Markov d'ordre 1.

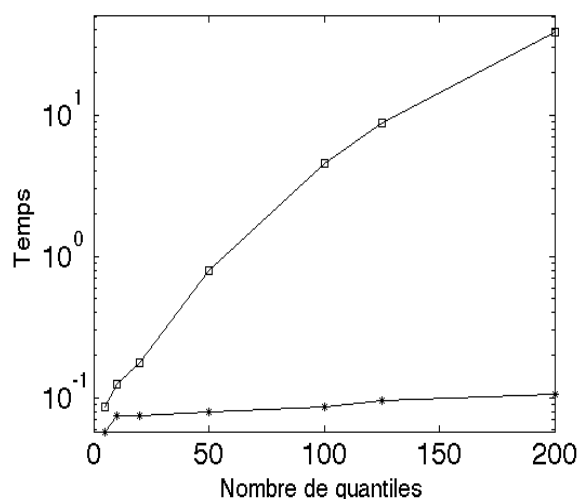


FIGURE 2.29 – Temps d'exécution de la méthode pour différents nombres de quantiles. Les ronds représentent les temps lorsque nous travaillons avec des chaînes de Markov d'ordre 1, les carrés sont pour l'ordre 2.

2.3 Conclusion

Dans ce chapitre, nous avons analysé une méthode permettant de créer un pont entre séries temporelles et réseaux. Plus précisément, nous pouvons analyser des séries temporelles par des mesures de graphes. À l'inverse, il est possible d'analyser un réseau grâce à des mesures statistiques de la série temporelle associée. Par exemple, les motifs dans un réseau sont associés aux périodicités dans un signal, caractérisées en s'intéressant au spectre de puissance du signal. Nous avons également montré que cette méthode pouvait être étendue en considérant des processus de Markov d'ordre supérieur ; la précision des constructions est améliorée.

Nous nous sommes principalement concentrés sur le fait de reconstruire une série temporelle à partir d'un réseau pour qu'elle soit la plus semblable à la série de base. En revanche, nous n'avons pas vraiment analysé comment les propriétés des séries (ou réseaux) se retrouvaient dans les réseaux (ou séries) associé(e)s ; la façon dont ces propriétés sont transférées est encore assez sibylline et mérite certainement d'être investiguée.

Chapitre 3

Graphes de visibilité

Dans le chapitre précédent, nous avons présenté une première méthode permettant de transformer une série temporelle en réseau et inversement. Nous continuons avec ce chapitre en présentant une seconde méthode permettant de transformer une série temporelle en un réseau afin d'étudier certaines de ses propriétés. Celle-ci est basée sur un critère purement géométrique et contrairement à la précédente, n'admet pas de pseudo-inverse. Néanmoins, cette méthode permet d'apercevoir plus facilement le lien entre les séries temporelles et les graphes associés. Pour cela, nous exposons très brièvement quelques résultats permettant de faire ce lien. La référence principale pour ce chapitre est [41].

3.1 Aspects théoriques

Nous ne considérons que les séries temporelles discrètes à une dimension. La série est vue comme un "paysage". Supposons une série temporelle $\{x(t_i)\}_{i=1\dots N}$ composée de N données. Les graphes de visibilité sont une manière de représenter une série temporelle en graphe où chaque point correspond à un nœud dans le graphe de visibilité associé. Cette représentation a été utilisée pour décrire et comparer théoriquement différents types de dynamiques [27, 42, 29] et a été appliquée pour caractériser différents signaux empiriques en extrayant certaines caractéristiques topologiques des graphes associés [3].

3.1.1 Graphe de visibilité naturelle

Pour l'algorithme de visibilité naturelle, deux nœuds sont reliés entre eux dans le graphe si nous pouvons relier leur point correspondant dans la série sans intersecter d'autres points (figure 3.1). En d'autres mots, deux points $x(t_i)$ et $x(t_j)$ dans la série sont en "visibilité", et donc les nœuds associées dans le graphe seront reliés, si pour n'importe quel point $x(t_k)$ avec $t_i < t_k < t_j$, le critère suivant est respecté

$$x(t_k) < x(t_i) + (x(t_j) - x(t_i)) \frac{t_k - t_i}{t_j - t_i}. \quad (3.1)$$

Il s'ensuit que le graphe sera toujours

- connecté : chaque nœud voit au moins ses plus proches voisins donc il est impossible d'avoir des nœuds isolés ;
- non dirigé ;
- invariant sous transformations affines de la série temporelle : le critère de visibilité ainsi défini n'est impacté ni par translation, ni par renormalisation horizontale ou verticale.

Bien sûr, nous perdons irrémédiablement de l'information lors de cette procédure. Par exemple, deux séries ayant la même période mais possédant des valeurs différentes $T1 = \dots, 2, 1, 2, 1, \dots$ et $T2 = \dots, 5, 2, 5, 2, \dots$ sont envoyées sur le même graphe de visibilité.

3.1.2 Graphe de visibilité horizontale

Pour l'algorithme de visibilité horizontale, deux nœuds i et j sont connectés dans le graphe si nous pouvons tracer, dans la série temporelle, un segment horizontal dont les extrémités sont $x(t_i)$ et $x(t_j)$ qui n'intersecte aucune donnée intermédiaire. Dit autrement, deux nœuds i et j sont connectés si

$$x(t_i), x(t_j) > x(t_n) \quad \forall t_n, \quad t_i < t_n < t_j. \quad (3.2)$$

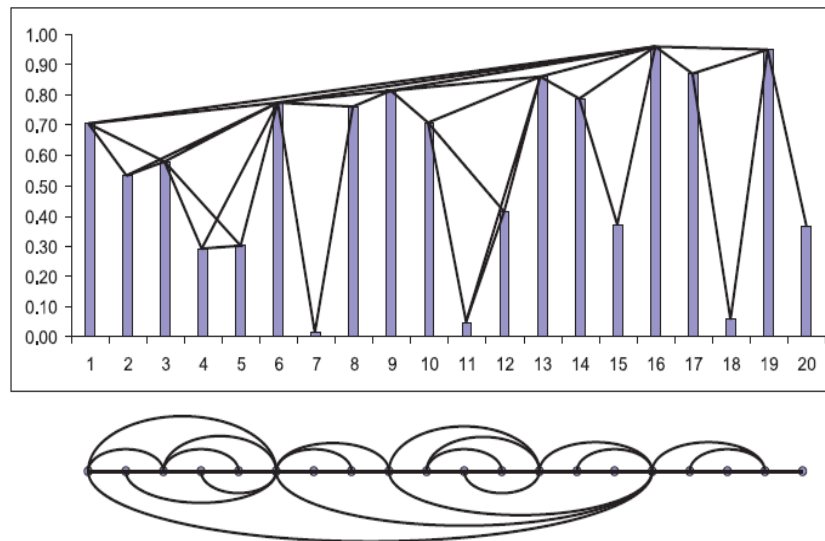


FIGURE 3.1 – Exemple d'un graphe de visibilité naturelle (NVg). Une série périodique est représentée sur la partie supérieure et le graphe de visibilité naturelle associé est représenté en dessous. Chaque donnée de la série représente un nœud dans le NVg et deux nœuds sont reliés si les données associées respectent l'éq. (3.1) [27].

À l'instar de la visibilité naturelle, le graphe résultant est connecté, non dirigé (cependant nous présentons dans la suite une version dirigée), invariant sous transformations affines de la série temporelle et nous fait perdre de l'information. La figure 3.2 illustre cet algorithme appliqué sur la même série qu'à la figure 3.1. Nous remarquons qu'il s'agit d'une simplification de l'algorithme de visibilité naturelle.

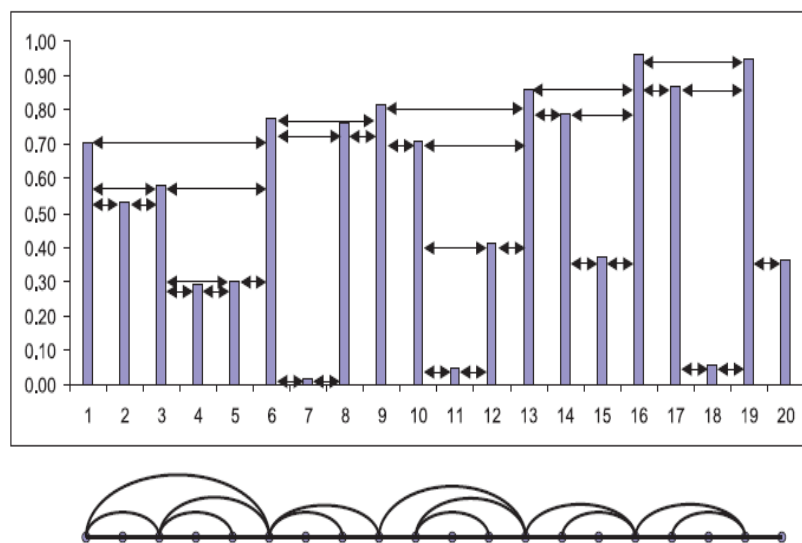


FIGURE 3.2 – Exemple d'un graphe de visibilité horizontale (HVg). Une série périodique est représentée sur la partie supérieure et le graphe de visibilité horizontale associé est représenté en dessous. Chaque donnée de la série représente un nœud dans le HVg et deux nœuds sont reliés si les données associées respectent l'éq. (3.2) [27].

Théorème 3.1.1. *Le degré moyen d'un graphe de visibilité horizontale (HVg) obtenu à partir d'une série périodique infinie de période T (sans répétition de mêmes valeurs dans la période) est*

$$\bar{k} = 4 \left(1 - \frac{1}{2T} \right). \quad (3.3)$$

Preuve. Soit la série $\{\dots x_{-1}, x_0, x_1, x_2, \dots, x_T, \dots\}$ de période T . Sans perte de généralité, x_0 et x_T sont les données ayant les plus grandes valeurs. Le HVg associé à cette série est en fait une concaténation de plusieurs motifs identiques correspondant aux sous-séries x_0, \dots, x_T . Chaque motif est donc composé de $T + 1$ nœuds et le degré moyen du HVg

correspond au degré moyen des motifs possédant T nœuds. En effet, pour les nœuds associés à x_0 et x_T , dans le motif, nous ne prenons en compte que la moitié de leur degré réel. Considérons à présent un motif et supposons qu'il est composé de V liens. Soit x_i la plus petite valeur de la sous-série (x_i est toujours bien définie car il n'y a pas de doublon au sein d'une période). Par construction, le nœud associé à x_i ne possède que deux liens et a donc un degré de 2. Retirons à présent ce nœud ainsi que ses liens ; le motif résultant ne contient plus que T nœuds et $V - 2$ liens (voir figure 3.3 pour une illustration du procédé). En itérant cette opération $T - 1$ fois, nous enlevons donc $2(T - 1)$ liens. Il en résulte un motif composé de deux nœuds (associés à x_0 et x_T) et un lien entre ceux-ci. Dès lors, le degré moyen \bar{k} est

$$\bar{k} \equiv 2 \frac{\text{\#liens}}{\text{\#nœuds}} = 2 \frac{2(T-1)+1}{T} = 4 \left(1 - \frac{1}{2T}\right).$$

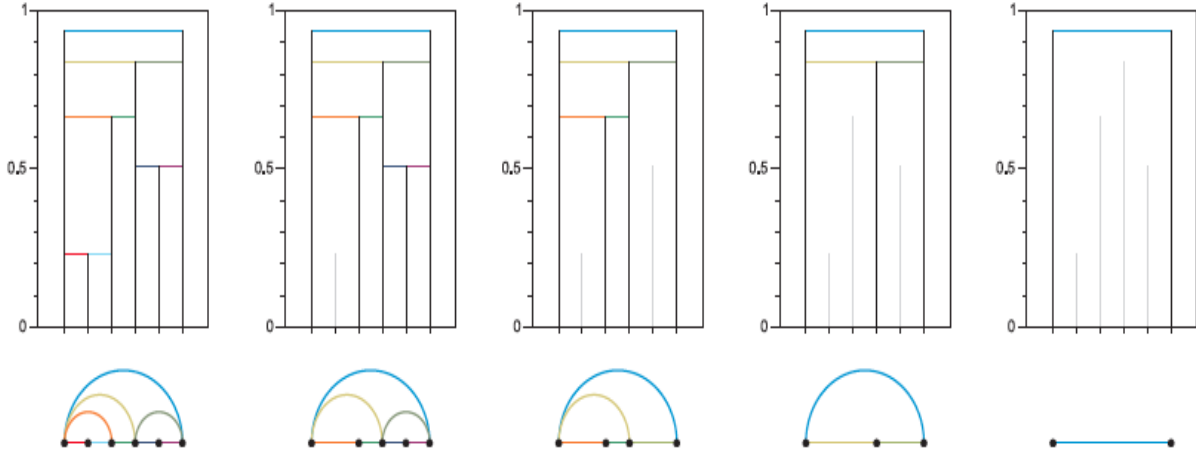


FIGURE 3.3 – Illustration du procédé pour la construction de la preuve du théorème 3.1.1 [42].

□

Ce résultat suggère également que le degré moyen de n'importe quel HVg issu d'une série d'un système dynamique est tel que $2 \leq \bar{k} \leq 4$. La borne inférieure est atteinte lorsque la série est constante puisque dans ce cas $T = 1$ tandis que la borne supérieure est atteinte pour des séries non périodiques, $T \rightarrow \infty$.

Théorème 3.1.2. Soit $\{x_i\}$ une série de variables aléatoires indépendantes et identiquement distribuées dont la probabilité de densité continue est $f(x)$. La distribution de degrés d'un HVg associé à une telle série est donnée par

$$P(k) = \frac{1}{3} \left(\frac{2}{3}\right)^{k-2}. \quad (3.4)$$

Preuve. Soit x une donnée d'une série telle que mentionnée dans l'énoncé. Soit Φ_2 la probabilité que la visibilité de cette donnée soit interrompue par une donnée x_g à gauche et x_d à droite. Cette probabilité est donnée par

$$\begin{aligned} \Phi_2 &= \int_{-\infty}^{\infty} \int_x^{\infty} \int_x^{\infty} f(x) f(x_g) f(x_d) dx dx_g dx_d \\ &= \int_{-\infty}^{\infty} f(x) \left(\int_x^{\infty} f(x_g) dx_g \right)^2 dx \\ &= \int_{-\infty}^{\infty} f(x) \left(1 - \int_{-\infty}^x f(x_g) dx_g \right)^2 dx \\ &= \int_{-\infty}^{\infty} f(x) (1 - F(x))^2 dx \\ &= \frac{1}{3} \end{aligned}$$

(intégration par partie) avec $F(x)$ la fonction de répartition définie par $F(x) = \int_{-\infty}^x f(x') dx'$. Soit $Q(k)$ la probabilité qu'une donnée en voie au moins k autres. Nous pouvons la définir de façon récursive de la façon suivante

$$Q(k) = Q(k-1)(1 - \Phi_2) = \frac{2}{3} Q(k-1).$$

Par induction, nous obtenons

$$Q(k) = \left(\frac{2}{3}\right)^{k-2}.$$

Dès lors, nous pouvons exprimer la probabilité $P(k)$ qu'une donnée voie exactement k autres données par

$$P(k) = Q(k)\Phi_2 = \frac{1}{3} \left(\frac{2}{3}\right)^{k-2}.$$

□

Grâce à ce résultat, nous pouvons calculer le degré moyen \bar{k} d'un HVg associé à une série aléatoire

$$\bar{k} = \sum kP(k) = \sum_{k=2}^{\infty} k \frac{1}{3} \left(\frac{2}{3}\right)^{k-2} = \frac{1}{3} \sum_{k=0}^{\infty} k \left(\frac{2}{3}\right)^k + \frac{2}{3} \sum_{k=0}^{\infty} \left(\frac{2}{3}\right)^k = \frac{1}{3} \cdot \frac{2/3}{(1-2/3)^2} + \frac{2}{3} \cdot \frac{1}{1-2/3} = 4$$

ce qui est bien en accord avec le résultat de l'éq.(3.3).

3.1.3 Graphe de visibilité horizontale dirigé

Jusqu'ici, nous n'avons considéré que des graphes non dirigés. Nous envisageons à présent des *directed horizontal visibility graphs* (DHVG) à savoir des *horizontal visibility graphs* où nous distinguons le degré entrant d'un nœud $k_{in}(x_i)$ du degré sortant $k_{out}(x_i)$ avec $k(x_i) = k_{in}(x_i) + k_{out}(x_i)$, le degré d'un nœud. Le degré entrant $k_{in}(x_i)$ d'un nœud associé à x_i est défini comme étant le nombre de liens qu'à ce nœud avec des nœuds "précédents" i.e., des nœuds associés aux données précédentes à x_i dans la série temporelle. À l'inverse, le degré sortant $k_{out}(x_i)$ est défini comme étant le nombre de liens qu'à ce nœud avec des nœuds associés à des données futures par rapport à x_i (figure 3.4).

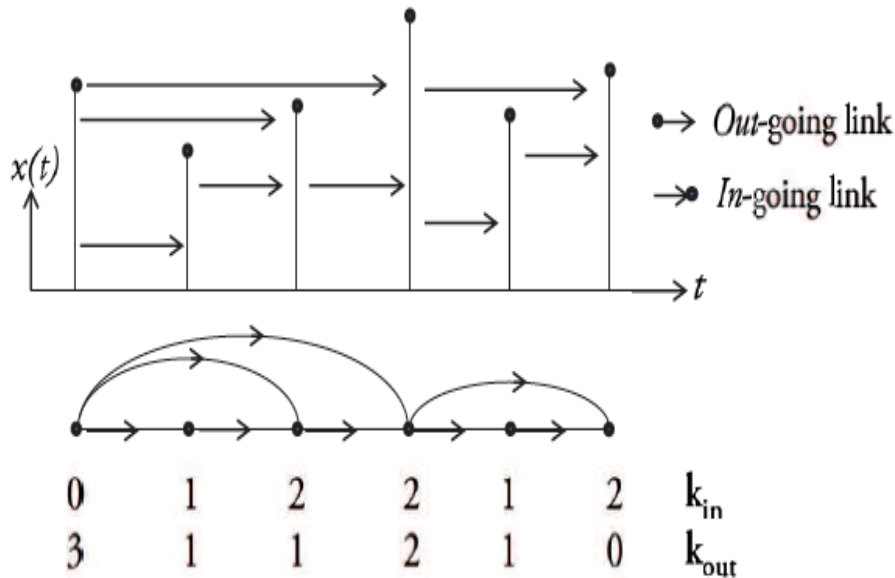


FIGURE 3.4 – Illustration de la méthode. Chaque donnée de la série $\{x_t\}$ représente un nœud dans le graphe. Une flèche entre deux données signifie que les nœuds associés sont reliés dans le DHVG associé. Chaque nœud possède un degré entrant k_{in} qui correspond au nombre de liens avec les nœuds "passés", et un degré sortant k_{out} correspondant au nombre de liens avec les nœuds futurs [27].

La distribution de degrés d'un graphe décrit la probabilité qu'un nœud ait un degré k i.e., k liens. La distribution de degrés entrants d'un DHVG est la distribution de probabilité de k_{in} , notée P_{in} , et est définie par $P_{in}(k) = P(k_{in} = k)$. De façon analogue, la distribution de degrés sortants d'un DHVG est la distribution de probabilité de k_{out} , notée P_{out} , et est définie par $P_{out}(k) = P(k_{out} = k)$.

Théorème 3.1.3. Soit $\{x_t\}_{t=-\infty, \dots, \infty}$ une série bi-infinie de variables aléatoires indépendantes et identiquement distribuées dont la probabilité de densité continue est $f(x)$. Alors, les distributions des degrés entrants et sortants du graphe de visibilité horizontale dirigé associé sont

$$P_{in}(k) = P_{out}(k) = \left(\frac{1}{2}\right)^k \quad k = 1, 2, 3, \dots$$

Preuve. (Distribution de degrés entrants) Soit x une donnée d'une série telle qu'il vient d'être mentionné. La probabilité

que la visibilité horizontale de cette donnée soit interrompue par une donnée à sa gauche x_g s'exprime comme

$$\begin{aligned}\Phi_1 &= \int_{-\infty}^{\infty} \int_x^{\infty} f(x)f(x_g)dx_gdx \\ &= \int_{-\infty}^{\infty} f(x)(1-F(x))dx \\ &= \frac{1}{2}\end{aligned}$$

(intégration par partie) avec $F(x)$ la fonction de répartition définie par $F(x) = \int_{-\infty}^x f(x')dx'$. Soit $Q(k)$ la probabilité qu'une donnée en voie au moins k autres. Nous pouvons la définir récursivement de la façon suivante

$$Q(k) = Q(k-1)(1-\Phi_1) = \frac{1}{2}Q(k-1).$$

Par induction, avec $Q(1) = 1$, nous obtenons

$$Q(k) = \left(\frac{1}{2}\right)^{k-1}.$$

Dès lors, nous pouvons exprimer la probabilité $P(k)$ qu'une donnée voie exactement k autres données par

$$P(k) = Q(k)\Phi_1 = \frac{1}{2}\left(\frac{1}{2}\right)^{k-1} = \left(\frac{1}{2}\right)^k.$$

La démarche est la même pour la distribution de degrés sortants. □

3.2 Quelques résultats

Après avoir défini les graphes de visibilité naturelle, horizontale et horizontale dirigé associés à différentes séries temporelles, nous nous intéressons, à travers quelques exemples, à la dualité entre ces graphes et séries.

3.2.1 Estimation de l'exposant de Hurst avec NVg

Le mouvement brownien fractionnaire (fBm) peut être utilisé comme cadre théorique dans l'étude des séries en temps réel apparaissant dans divers domaines. À titre d'exemples, les fBms sont utilisés pour caractériser des signaux physiologiques tels que les rythmes cardiaques [4], quantifier des corrélations dans les séquences d'ADN de base [46] ou encore décrire des données géologiques [21].

Un mouvement brownien fractionnaire est un processus aléatoire non stationnaire auto-similaire avec des incréments stationnaires (bruit gaussien fractionnaire, fGn) qui peuvent être caractérisés par l'exposant de Hurst H [28, 38]. En d'autres mots, le fBm est une généralisation du mouvement brownien. Un mouvement brownien permet de décrire mathématiquement le mouvement aléatoire d'une "grosse" particule plongée dans un fluide et qui n'est soumise qu'aux interactions produites par des chocs avec les "petites" molécules du fluide environnant [63]. Un fBm défini sur $[0, T]$, noté $B_H(t)$, est un processus Gaussien continu, commençant à zéro, de moyenne nulle pour tout $t \in [0, T]$ et de covariance

$$E(B_H(s)B_H(t)) = \frac{1}{2}(|s|^{2H} + |t|^{2H} - |s-t|^{2H})$$

avec H un réel dans $[0,1]$ appelé exposant de Hurst. Ce dernier permet de décrire la rugosité du mouvement résultant ; au plus cette valeur est petite, au plus le mouvement est "rugueux". En effet, l'exposant de Hurst détermine le genre de processus qu'est le fBm : un processus brownien est obtenu lorsque $H = 1/2$, tandis que pour $H > 1/2$ le signal montre des persistances (mathématiquement $\sum_{n=1}^{\infty} E(B_H(1)(B_H(n+1) - B_H(n))) = \infty$) ou encore les incréments du processus sont positivement corrélés alors que pour $H < 1/2$, le signal montre des anti-persistances ou encore les incréments du processus sont négativement corrélés (figure 3.5). Les incréments du processus sont en fait les bruits Gaussiens fractionnaires et sont définis comme $X(t) = B_H(t+1) - B_H(t)$. Comme dit précédemment, le fBm est un processus auto-similaire

$$\forall a > 0, B_H(at) \sim |a|^H B_H(t)$$

et les incréments sont stationnaires

$$B_H(t+s) - B_H(s) \sim B_H(t).$$

Il existe de nombreuses manières de générer et d'estimer des fBms [68, 14, 20, 12, 24]. Malheureusement, il est difficile de déterminer une méthode optimale. Cela vient du fait que le formalisme de fBm est exact uniquement lorsque la série toute entière (donc infinie) est considérée. Bien entendu, en pratique, seules les séries finies sont envisagées. Dès lors, les corrélations à longue portée en sont impactées et les dynamiques locales correspondant à une fenêtre de temps particulière sont surestimées.

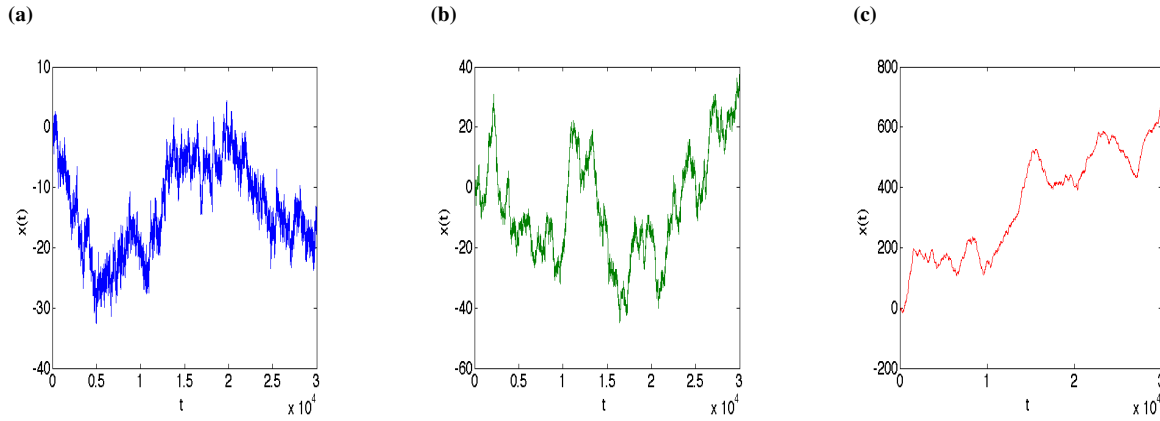


FIGURE 3.5 – Exemples de fBms avec un exposant de Hurst H égal à (a) 0.3 ; (b) 0.5 ; (c) 0.8. Plus l'exposant de Hurst est élevé, au plus le signal est lisse.

Dans cette section, nous proposons d'utiliser les NVgs pour estimer l'exposant de Hurst de séries fBm [28]. Lorsque nous calculons le graphe NVg associé à une série temporelle fBm, nous obtenons un graphe invariant d'échelle (*scale-free*, figure 3.6). Un tel graphe est caractérisé par le fait que les degrés des nœuds suivent une loi de puissance : la proportion des nœuds ayant un degré k est proportionnelle à $k^{-\gamma}$ lorsque k est suffisamment grand et l'exposant d'invariance d'échelle γ est strictement positif. Par exemple, le réseau du web ou encore les réseaux sociaux semblent se comporter comme des réseaux invariants d'échelle, d'où l'importance de ce type de réseau [57].

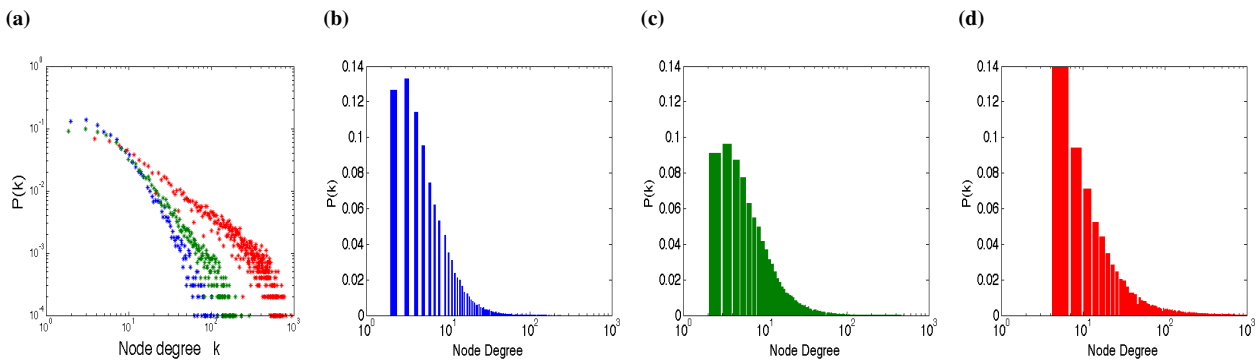


FIGURE 3.6 – (a) Distribution de degrés de différents NVgs associés à des séries fBm de 30 000 données avec en - bleu : $H = 0.3$, (anti-persistant) vert : $H = 0.5$ (mouvement Brownien sans mémoire), rouge : $H = 0.8$ (persistant). Les trois HVg sont *scale-free* puisque leur distribution de degrés suit une loi de puissance $P(k) = k^{-\gamma}$ avec $\gamma_{0.3} > \gamma_{0.5} > \gamma_{0.8}$. Afin de mieux observer les distributions *power law*, les panels (b)-(d) représentent pour chacun des fBms la distribution en bins log.

Le fait qu'un fBm donne un graphe invariant d'échelle n'est pas si surprenant. En effet, les hubs i.e. les nœuds les plus connectés, sont responsables de la structure en "triangle" de la queue de la distribution de degrés. Ces nœuds représentent en réalité les valeurs les plus grandes de la série puisque, de façon évidente, une donnée ayant une grande valeur (par rapport aux autres valeurs de la série) possède une grande connectivité (éq. (3.1)).

Pour calculer la queue de la distribution, nous nous focalisons principalement sur les hubs et calculons la probabilité que possède un nœud associé une valeur extrême d'avoir un degré k . Pour cela, supposons que la donnée de la série au temps t soit une valeur extrême (hub) $B_H(t) = h$. La probabilité qu'à ce hub d'avoir un degré T est donnée par

$$P(T) \sim P_{fr}(T)r(T). \quad (3.5)$$

$P_{fr}(T)$ correspond à la probabilité qu'après un temps T , la série atteigne la même valeur extrême i.e. $B_H(t+T) = h$. En outre, la visibilité en t est interrompue en $t+T$. En fait, $P_{fr}(T)$ n'est rien d'autre que la *first return time distribution*. Il a été démontré dans [15] que pour des séries fBm, $P_{fr}(T) \sim T^{H-2}$. Quant à $r(T)$, il définit le pourcentage de nœuds entre t et $t+T$ que t peut voir. Nous devons considérer la rugosité de la série pour définir le pourcentage de nœuds visibles entre deux valeurs extrêmes : plus la série est "rugueuse", au plus le pourcentage sera faible. En fait, la rugosité de la série est encodée dans son écart-type qui se comporte, pour fBm, comme T^H [35]. En normalisant, nous avons $r(T) \sim T^H/T$. En remarquant que, dans ce contexte, $T \equiv k$, nous avons, via l'éq. (3.5)

$$P(k) \sim k^{H-2}k^H/k = k^{2H-3}. \quad (3.6)$$

Cette dernière permet de relier l'exposant de Hurst H à l'exposant d'invariance d'échelle γ par la relation

$$\gamma(H) = 3 - 2H. \quad (3.7)$$

Nous vérifions "empiriquement" en calculant l'exposant de différents *scale-free* NVgs associé à des séries fBms artificielles ($H \in [0,1]$) de 30 000 données. Ces différentes séries sont générées au moyen d'un algorithme basé sur les ondelettes (*wfbm*, [1, 5]). Malheureusement, les générateurs de telles séries ne fournissent pas des résultats totalement exacts et nous avons donc inévitablement quelques biais et par conséquent, les exposants de Hurst H voulus ont des erreurs. Pour chaque valeur d'exposant H , nous avons généré 10 réalisations de fBms pour ainsi obtenir une estimation de la valeur de γ telle que $P(k) \sim k^{-\gamma}$. Cette estimation est obtenue par l'estimation du maximum de vraisemblance [39]. Nos degrés suivent une loi $P(k) \sim k^{-\gamma}$ à partir d'un certain degré minimal k_{min} . Autrement dit $P(k) = Ck^{-\gamma}$ avec C une constante pour tous les $k > k_{min}$. Nous voulons donc estimer l'exposant γ (pour les $k > k_{min}$). Nous pouvons normaliser la constante C par la contrainte

$$1 = \int_{k_{min}}^{\infty} p(k) dk = \int_{k_{min}}^{\infty} Ck^{-\gamma} dk = \frac{C}{1-\gamma} [k^{-\gamma+1}]_{k_{min}}^{\infty}. \quad (3.8)$$

Remarquons que cette expression n'a de sens que si $\gamma > 1$ puisque sinon $[k^{-\gamma+1}]_{k_{min}}^{\infty}$ diverge. En conséquence, les *power-law* dont l'exposant est inférieur à 1 ne peuvent être normalisés. Dès lors, si $\gamma > 1$, l'éq. (3.8) donne

$$C = \frac{\gamma-1}{k_{min}^{-\gamma+1}}.$$

Nous pouvons dès lors exprimer la distribution de degrés comme

$$P(k) = \frac{\gamma-1}{k_{min}} \left(\frac{k}{k_{min}} \right)^{-\gamma}.$$

Soit un ensemble de valeurs k_i ($i=1, \dots, n$) donné, la probabilité que ces valeurs soient générées par cette distribution $P(k)$ est proportionnelle à

$$P(k|\gamma) = \prod_{i=1}^n p(k_i) = \prod_{i=1}^n \frac{\gamma-1}{k_{min}} \left(\frac{k_i}{k_{min}} \right)^{-\gamma}. \quad (3.9)$$

Pour trouver la valeur de γ , nous avons besoin de connaître la probabilité $P(\gamma|k)$ qui donne la probabilité d'une valeur particulière γ étant donné les observations k_i . Par la formule de Bayes, nous avons

$$P(\gamma|k) = P(k|\gamma) \frac{P(\gamma)}{P(k)}. \quad (3.10)$$

La probabilité $P(k)$ est fixée puisque k est fixe. En effet, k correspond à un ensemble de données observées (ici, les degrés observés) qui ne varient pas. En outre, en l'absence d'informations particulières, nous supposons que la probabilité de l'exposant $P(\gamma)$ est uniforme i.e., une constante indépendante de γ . Dès lors, par l'éq. (3.10), nous avons $P(\gamma|k) \sim P(k|\gamma)$. En prenant le logarithme de $P(\gamma|k)$, nous avons

$$\begin{aligned} \mathcal{L} &= \ln P(\gamma|k) \\ &= \sum_{i=1}^n \left(\ln(\gamma-1) - \ln k_{min} - \gamma \ln \frac{k_i}{k_{min}} \right) \\ &= n \ln(\gamma-1) - n \ln k_{min} - \gamma \sum_{i=1}^n \ln \frac{k_i}{k_{min}}. \end{aligned} \quad (3.11)$$

En utilisant la méthode du maximum de vraisemblance, nous calculons la valeur la plus probable pour γ . En posant $\partial \mathcal{L} / \partial \gamma = 0$, nous trouvons

$$0 = \frac{n}{\gamma-1} - \sum_{i=1}^n \ln \frac{k_i}{k_{min}}$$

ou encore

$$\gamma = 1 + n \left(\sum_{i=1}^n \log \frac{k_i}{k_{min}} \right)^{-1}. \quad (3.12)$$

C'est cette dernière expression que nous utilisons dans la suite avec $k_{min} = 15$. Bien entendu, il ne s'agit que d'une approximation de l'exposant γ . Une estimation de l'erreur produite est donnée par [39]

$$\sigma = \frac{\gamma-1}{\sqrt{n}}.$$

Nous traçons la relation obtenue à la figure 3.7 : en bleu sont représentés la moyenne des résultats calculés (par l'éq. (3.12)) ainsi que les "variations", la ligne noire représente la meilleure droite de régression linéaire obtenue : $\gamma(H) = 3.09(\pm 0.1) - 2.06(\pm 0.1)H$, et la ligne magenta représente la relation théorique $\gamma(H) = 3 - 2H$ (eq. (3.7)). Bien que nous ayons quelques déviations par rapport à la relation théorique, les résultats obtenus sont globalement en accord avec la relation théorique. Nous remarquons qu'au plus nous nous approchons de $H = 0.5$, au plus les résultats sont proches de la théorie.

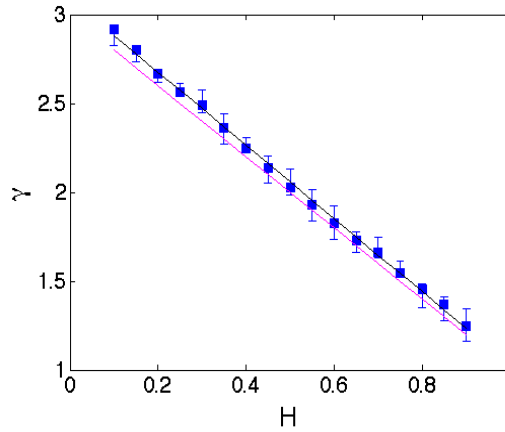


FIGURE 3.7 – Estimation de l'exposant γ de la distribution de degrés $P(k) \sim k^{-\gamma}$ des NVgs associés à différentes séries fBms caractérisées par l'exposant $H \in [0.1 ; 0.9]$. En bleu, la moyennes des résultats (carrés) ainsi que les barres d'erreur sur base 10 réalisations, la ligne noire représente la meilleure droite de régression linéaire obtenue (pour la moyenne des résultats) : $\gamma(H) = 3.09(\pm 0.1) - 2.06(\pm 0.1)H$, et la ligne magenta représente la relation théorique $\gamma(H) = 3 - 2H$ (éq. (3.7)).

En vue de conforter la consistance de l'algorithme de visibilité, nous procédons à une estimation du spectre de puissance. Les fBms possèdent un spectre de puissance qui se comporte comme $1/f^\beta$, avec β relié à l'exposant de Hurst par la relation [2]

$$\beta(H) = 1 + 2H. \quad (3.13)$$

En combinant les équations (3.12) et (3.13), les NVgs associés aux séries avec $1/f^\beta$ -noise (i.e. avec un spectre de puissance qui se comporte comme $1/f^\beta$) possèdent une distribution de degrés $P(k) \sim k^{-\gamma}$ avec γ vérifiant la relation

$$\gamma(\beta) = 4 - \beta. \quad (3.14)$$

La figure 3.8 reprend les résultats empiriques (en bleu) obtenus pour différentes séries $1/f^\beta$ artificiellement générées. Nous faisons varier β entre 1 et 2.5 et pour chacune de ces valeurs, nous générons 10 séries $1/f^\beta$. Encore une fois, nous avons estimé γ par l'éq. (3.12), avec $k_{min} = 15$. La ligne noire correspond à la meilleure droite de régression linéaire obtenue $\gamma(\beta) = 4.05(\pm 0.1) - 1.03(\pm 0.1)\beta$ et la ligne magenta représente le résultat théorique (éq. 3.14).

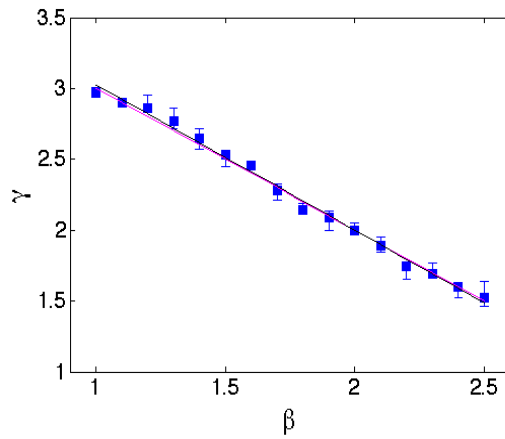


FIGURE 3.8 – Estimation de l'exposant γ de la distribution de degrés $P(k) \sim k^{-\gamma}$ des NVgs associés à différentes séries $1/f^\beta$ caractérisées par $\beta \in [1 ; 2.5]$. En bleu, la moyennes des résultats (carrés) ainsi que les barres d'erreur sur base 10 réalisations, la ligne noire représente la meilleure droite de régression linéaire obtenue : $\gamma(H) = 4.05(\pm 0.1) - 1.03(\pm 0.1)\beta$, et la ligne magenta représente la relation théorique $\gamma(\beta) = 4 - \beta$ (éq. (3.14)).

3.2.2 Filtrage de bruit avec HVg

Dans cette section, nous abordons le problème de filtrer un signal périodique bruité. Il existe de nombreux algorithmes permettant de détecter des périodes dans un signal [45] souvent basés sur l'autocorrélation (domaine temporel) et l'analyse spectrale (domaine fréquentiel). Ici, nous utilisons les HVgs.

Si nous superposons un bruit sur une série périodique (bruit *extrinsèque*), nous pouvons nous attendre à ce que le degré des nœuds associés aux petites valeurs de la série reste constant, ou du moins varie très peu. À l'inverse, les

nœuds correspondants aux valeurs élevées vont éventuellement augmenter puisque ces points sont susceptibles d'avoir une meilleure visibilité. Dès lors, la structure en forme de delta de la queue de la distribution de degrés va être perturbée et une distribution exponentielle apparaîtra en raison de la présence d'un tel bruit. L'algorithme est capable de détecter de telles séries car la distribution de degrés peut être calculée analytiquement.

Considérons, par simplicité, une série périodique $\{x_i\}$ de période $T = 2$ à laquelle est ajoutée une petite quantité de bruit (figure 3.9). Les données de la série peuvent être séparées en deux groupes : $\{x_1, x_3, x_5, \dots\}$ dont les valeurs sont petites et $\{x_2, x_4, x_6, \dots\}$ dont les valeurs sont grandes. Par construction, les nœuds associés aux valeurs impaires ne voient que leurs plus proches voisins ; ils ont un degré k égal à 2. Considérons à présent uniquement les nœuds associés aux valeurs paires. Ces nœuds forment un sous-graphe se réduisant essentiellement à un graphe associé à une série non corrélée dont la distribution de degrés est donnée par l'éq. (3.4). Dès lors, en considérant le graphe entier, nous obtenons la distribution de degrés suivante

$$\begin{aligned} P(2) &= 1/2 \\ P(3) &= 0 \\ P(k+2) &= \frac{1}{3} \left(\frac{2}{3} \right)^{k-2}. \end{aligned} \quad (3.15)$$

En d'autres mots, le fait d'ajouter une petite quantité de bruit à une série périodique génère une queue exponentielle dans la distribution de degrés du HVg correspondant. Cette distribution de degrés possède la même pente que celle associée à une série purement non corrélée. Le degré moyen \bar{k} s'exprime

$$\bar{k} = \sum_{k=2}^{\infty} kP(k) = 4,$$

ce qui suggère qu'il n'y a pas de périodicité.

Sur la figure 3.9, nous avons tracé la distribution de degrés d'une série périodique bruitée de période 2. Le bruit ajouté est un bruit gaussien, suivant une loi $N(0,0.05)$. Conformément à l'éq. (3.15), nous avons bien $P(2) = 1/2$ mais n'obtenons pas $P(3) = 0$. Cela est dû à la taille finie de la série. L'algorithme détecte les deux signaux et sait donc distinguer le bruit ajouté.

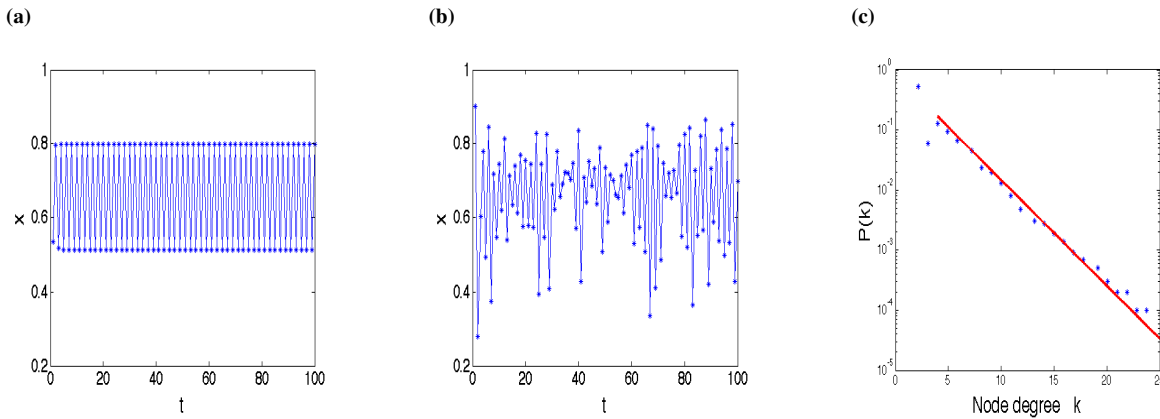


FIGURE 3.9 – (a) 100 premières itérations d'une série de 50 000 données. La série est définie par $x_{n+1} = \mu x_n (1 - x_n)$ avec $\mu = 3.2$ (puisque pour cette valeur, nous avons un comportement périodique de période $T = 2$) ; (b) Même série que sur le panel (a) à laquelle nous ajoutons un bruit suivant une loi normale $N(0,0.05)$; (c) Distribution de degrés : les points bleus représentent celle associée au HVg tandis que la ligne rouge est le résultat théorique (eq. (3.15)).

Définition et exemples

Soient $S = \{x_i\}_{i=1, \dots, n}$ une série périodique de période T avec $T \ll n$ et S_b cette même série mais polluée par un bruit. Sans perte de généralité, nous considérons un bruit issu d'une loi uniforme $U([-0.5, 0.5])$. Soit une filtre $f \in [\min(x_i), \max(x_i)]$. Le *filter horizontal visibility graph* f -HVg associé à une série S_b se construit de la façon suivante

1. chaque x_i représente un nœud dans le graphe f -HVg correspondant ;
2. deux nœuds i et j sont reliés dans le f -HVg si et seulement si

$$x_i, x_j > x_n + f \quad \forall n \text{ tel que } i < n < j.$$

Pour filtrer le bruit d'un signal périodique bruité S_b , nous générons les différents f -HVg associés pour différentes valeurs croissantes de f . Pour chaque f -HVg ainsi obtenu, le degré moyen \bar{k} est calculé. Pour un intervalle approprié $[f_{\min}, f_{\max}]$, le f -HVg associé à la série S_b sera équivalent au HVg associé à la série S , lequel à un degré moyen bien défini en fonction de la période de la série S (éq. (3.3)). Dans cet intervalle, le degré moyen sera constant et dès lors, par l'éq. 3.3, nous pourrions en déduire la période.

En vue d'illustrer cela, nous générons une série périodique de période $T = 2$ à laquelle nous ajoutons un bruit. Cela est illustré à la figure 3.10 ainsi que les résultats du filtrage où nous avons tracé les différentes valeurs de \bar{k} en fonction de f . Sans surprise, le degré moyen égale 4 lorsque $f = 0$ puisque dans ce cas, le f -HVg d'une série périodique bruitée correspond au HVg d'une série non périodique. Le degré moyen décroît (en fonction de f) jusqu'à atteindre 2, valeur minimale. Nous remarquons un plateau pour $\bar{k} = 3$, ce qui, par l'éq. (3.3), correspond à une période $T = (2 - \frac{\bar{k}}{2})^{-1} = 2$. Un second plateau est détecté pour $\bar{k} = 2$, lorsque le filtre f est assez grand pour que les données n'aient une visibilité que de leurs voisins.

À titre de comparaison, nous calculons également la fonction d'autocorrélation $ACF(\tau)$. Rappelons que cette dernière est périodique de période T lorsque la série est elle-même de période T . En revanche, lorsque la série est non périodique, la fonction d'autocorrélation ne présente aucune structure particulière. Dans notre cas, l'autocorrélation présente une structure de période $T = 2$, comme notre série de base.

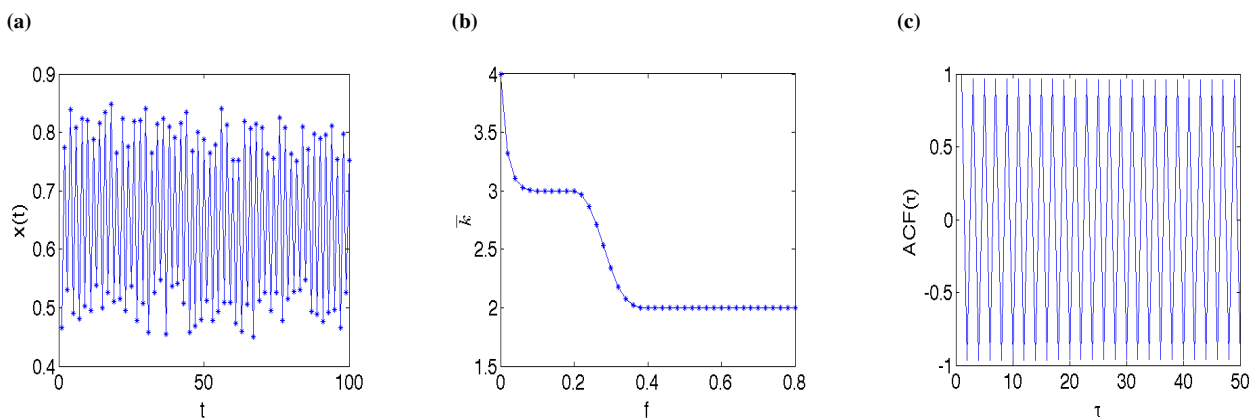


FIGURE 3.10 – (a) Série périodique de période $T = 2$ à laquelle un bruit est ajouté. Ce bruit suit une loi uniforme $U([-0.5, 0.5])$ d'amplitude 0.1 ; (b) Degré moyen \bar{k} en fonction du filtre f . Un premier plateau est détecté pour $\bar{k} = 3$, ce qui correspond à une période $T = 2$. Un second plateau apparaît pour $\bar{k} = 2$, lorsque f est suffisamment grand pour empêcher aux données de voir plus que leur voisins ; (c) La fonction d'autocorrélation de la série suggérant que cette dernière est de période 2.

Chaotique versus périodique-bruit

Considérons à présent le modèle logistique

$$x_{t+1} = \mu x_t (1 - x_t)$$

avec $x \in [0; 1]$ et $\mu \in [0; 4]$. L'attracteur chaotique est l'intervalle $[0, 1]$ entier uniquement pour $\mu = 4$. En particulier, pour $\mu \in [3.6, 3.67]$, l'attracteur est divisé en deux bandes entre lesquelles l'orbite chaotique voyage de l'une à l'autre. Par exemple, prenons la valeur $\mu = 3.65$. La figure 3.11 affiche la série composée de 10 000 points ainsi que les 100 première valeurs, pour plus de clarté. Dans cette situation, la série est chaotique et non périodique. Cependant, la fonction d'autocorrélation suggère la présence d'une périodicité dans la série (figure 3.11(d)) ce qui n'est pas sans rappeler la structure en deux bandes de l'attracteur. Néanmoins, lorsque nous appliquons la technique de filtrage, nous obtenons un résultat contraire : il n'y a aucun plateau détecté (figure 3.11(c)), ce qui suggère que la série n'est pas périodique. Lorsque nous disons qu'il n'y a pas de plateau, nous excluons celui présent pour $\bar{k} = 2$ car il correspond aux valeurs de f suffisamment grandes pour que les données n'aient une visibilité que de leurs voisins. Dès lors, nous pouvons considérer cette technique de filtrage comme un test supplémentaire (à la fonction d'autocorrélation) pour détecter ou non des périodicités.

Nous pouvons analyser la distribution de degrés du HVg correspondant (figure 3.12). Nous trouvons $P(2) = 1/2$, ce qui signifie que chaque bande (figure 3.11(a)) contient la moitié des données, ce qui est en accord avec le fait que l'orbite chaotique alterne entre les deux bandes. De plus, la distribution de degrés affiche une décroissance exponentielle dont la pente est moindre que celle d'un processus aléatoire non corrélé (éq. (3.4)).

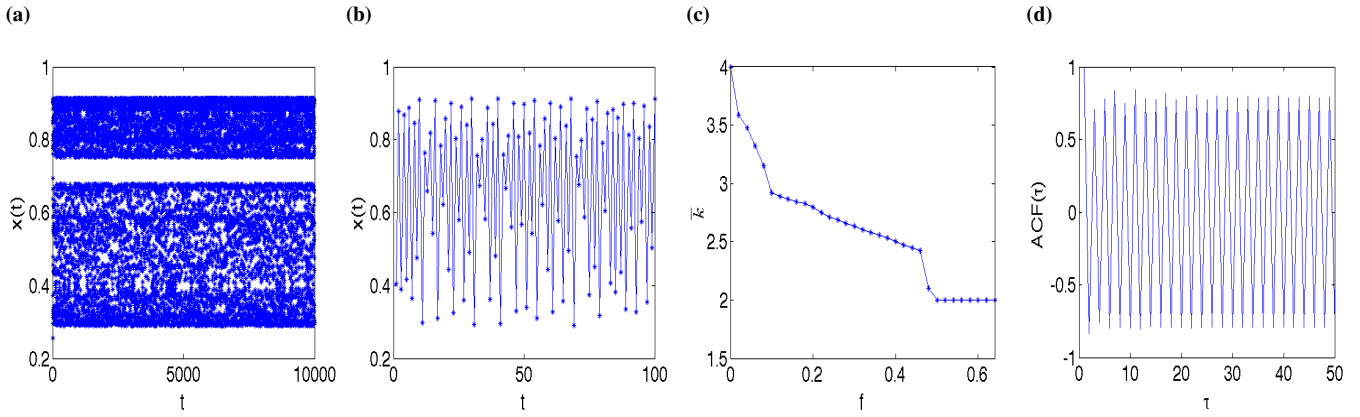


FIGURE 3.11 – (a) Série extraite de la logistic map avec $\mu = 3.65$; (b) Même série où seuls les 100 premiers points sont représentés ; (c) Degré moyen en fonction de f . Aucun seuil n'est trouvé suggérant l'absence de périodicité ; (d) Fonction d'autocorrélation de la série suggérant la présence d'une périodicité dans la série.

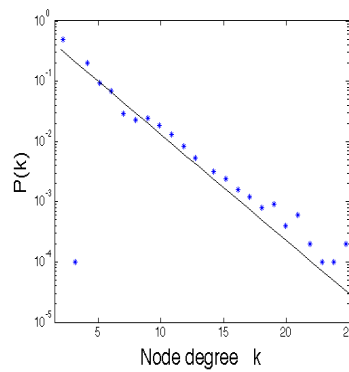


FIGURE 3.12 – Distribution de degrés $P(k)$ du HVg associé à la série chaotique de la figure 3.11. La ligne noire représente la distribution de degrés d'une série aléatoire non corrélée (eq. (3.4)).

3.2.3 Mesure de l'irréversibilité avec DHVg

Soit un processus stationnaire $\{x_1, x_2, \dots, x_n\}$. Ce dernier est statistiquement temps réversible (temps réversible par abus de langage) si, pour chaque n , les séries $\{x_1, x_2, \dots, x_n\}$ et $\{x_n, \dots, x_2, x_1\}$ possèdent la même distribution de probabilité conjointe [59]. Autrement dit, une série temporelle inversible et sa série temps inversé sont, d'un point de vue statistique, également probables. Les processus linéaires Gaussiens font parties des processus temps réversibles et sont associées aux processus en équilibre thermique en physique statistique. À l'inverse, les séries temps irréversibles indiquent la présence de non linéarités dans les dynamiques sous-jacentes (processus stochastiques non Gaussiens, chaos dissipatif) et sont associées aux processus mis hors équilibre en thermodynamique.

Nous utilisons l'algorithme de visibilité horizontale pour tester l'irréversibilité de différentes séries stochastiques stationnaires. En particulier, nous nous intéressons à la version dirigée de l'algorithme. De plus, nous utilisons la divergence de Kullback-Leibler (entropie relative) entre les distributions des degrés sortants et entrants : $D[P_{out}(k)||P_{in}(k)]$ [23].

Quantification de l'irréversibilité

Comme dit précédemment, nous utilisons les DHVgs. Cela est motivé par la supposition que l'information emmagasinée dans les distributions de degrés entrants et sortants prend en compte la quantité d'irréversibilité dans les séries associées. Plus précisément, nous pensons que cela peut être mesuré en calculant la distance (au sens des distributions) entre les distributions de degrés $P_{out}(k)$ et $P_{in}(k)$.

Pour calculer cette distance, nous utilisons la divergence de Kullback-Leibler (KLD) (ou entropie relative). Cette dernière, principalement utilisée en théorie des probabilités et en théorie de l'information, est une mesure de dissimilarité entre deux distributions de probabilités [64]. Soient une variable x et deux distributions de probabilités $p(x)$ et $q(x)$. La KLD entre ces dernières est définie comme

$$D[p||q] = \sum_x p(x) \log \frac{p(x)}{q(x)}.$$

$D[p||q]$ égale 0 uniquement lorsque $p(x) = q(x)$ et dans les autres cas, elle est positive : au plus p et q sont "distinguables", au plus la KLD entre ces deux distributions sera élevée.

Ici, pour calculer l'irréversibilité de différentes séries stochastiques stationnaires réelles mais à temps discret $\{x_t\}_{t=1,\dots,N}$, nous n'utilisons que les distributions de degrés sortants $P_{out}(k)$ et entrants $P_{in}(k)$. Nous comparons la distribution de degrés sortants de la série "forward" $P_{kout}(k|\{x_t\}_{t=1,\dots,N}) = P_{out}(k)$ avec la distribution de degrés sortants de la même série mais "backward" i.e. $P_{kout}(k|\{x_t\}_{t=N,\dots,1})$. Or cette dernière n'est rien d'autre que la distribution de degrés entrants de la série "forward" : $P_{kout}(k|\{x_t\}_{t=N,\dots,1}) = P_{in}(k)$. La KLD entre ces deux distributions est

$$D[P_{out}(k)||P_{in}(k)] = \sum_k P_{out}(k) \log \frac{P_{out}(k)}{P_{in}(k)}. \quad (3.16)$$

Encore une fois, cette expression s'annule uniquement lorsque $P_{out}(k) = P_{in}(k)$ et est positive autrement.

Réversibilité

Nous proposons de tester la méthode sur quelques séries temps-réversibles.

■ Séries stochastiques non corrélées $U([0,1])$

À la figure 3.13, nous avons repris la distribution de degrés *in* et *out* pour une série de 50 000 données. Nous ne pouvons faire de distinction entre les deux distributions et en conséquence, KLD est très proche de zéro (voire table 3.1 pour des valeurs numériques) ce qui indique que la série est temps-réversible. Les résultats obtenus à la figure 3.13 sont en accord avec le théorème 3.1.3, qui stipule que $P_{in}(k) = P_{out}(k) = (\frac{1}{2})^k$ avec $k = 1, 2, 3, \dots$. Subséquemment, la KLD entre les distributions de degrés sortants et entrants associées aux séries aléatoires non corrélées tend asymptotiquement vers zéro avec la taille des séries, ce qui suggère bien que les séries sont temps-réversibles.

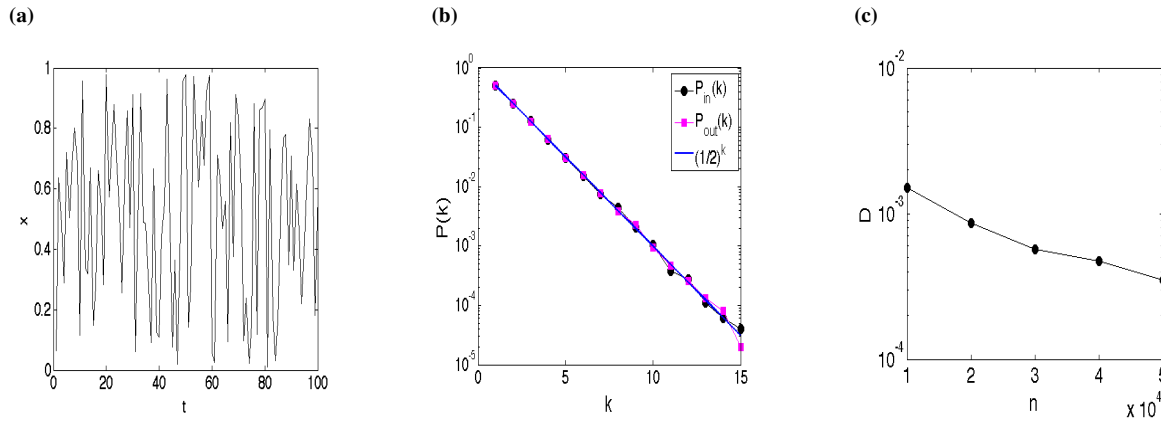


FIGURE 3.13 – (a) Série stochastique non corrélée réversible $U([0,1])$. ; (b) Distribution de degrés $P_{in}(k)$ et $P_{out}(k)$. Ces distributions sont en accord avec la relation théorique $P_{in}(k) = P_{out}(k) = (\frac{1}{2})^k$ avec $k = 1, 2, 3, \dots$ (théorème 3.1.3); (c) Divergence de Kullback-Leibler $D[P_{out}(k)||P_{in}(k)]$.

■ Séries stochastiques corrélées

En particulier, ces séries sont obtenues par *minimal subtraction procedure* [30]. Cette technique permet de générer des séries aléatoires corrélées Gaussiennes $\{x_i\}$ de moyenne nulle et de fonction de corrélation $\langle x_i, x_j \rangle = C(|i - j|)$. Cette méthode est une "amélioration" de la *Fourier filtering method* (FFM) en ce sens qu'elle permet d'éliminer les effets de coupure indésirables pour les fonctions de corrélation et prend en compte la nature discrète de la série. En quelques mots, la FFM procède en filtrant les composants de Fourier d'une série non corrélée de nombres aléatoires au moyen d'un filtre donné (souvent, une fonction *power-law*) dans le but d'introduire des corrélations dans les variables. Ici, la méthode utilisée se base sur le fait que n'importe quelle fonction $C(t)$ peut être considérée comme un fonction de corrélation (bien que quelques exigences doivent être respectées ; en particulier la forme quadratique $\sum_{ij} x_i C(|i - j|) x_j$ doit être définie positive). Par exemple, si nous souhaitons générer une série avec une fonction de corrélation se comportant asymptotiquement comme $C(t) \sim t^{-\gamma}$, nous avons un problème puisque cette fonction diverge lorsque $t \rightarrow 0$. Pour éviter cette singularité, les auteurs de [34] ont introduit la fonction de corrélation $C(t) = (1 + t^2)^{-\gamma/2}$. Cependant, si nous prenons cette fonction de corrélation dans notre cas, la transformée de Fourier $S(k) = N^{1/2} \sum_{j=1}^N \exp(ijk/N) C(j)$ prend des valeurs négatives pour certains k , ce qui n'est pas acceptable. Pour pallier à ce problème, nous définissons une nouvelle densité spectrale $S_0(k) = S(k) - S_{min}(k)$ avec $S_{min}(k)$ la valeur minimal de $S(k)$. Nous utilisons à présent $S_0(k)$ dans l'étape du filtre. L'unique effet qu'à cette *minimal subtraction procedure* est que $C(0)$ ne vaut plus 1 mais adopte la valeur minimale nécessaire pour que la forme quadratique soit définie positive. En résumé, nous procédons comme suit :

1. Générer un ensemble de variables Gaussiennes aléatoires de moyenne nulle et de variance égale à 1 : $\{u_j\}$, $j = 1, \dots, N$
2. Calculer sa transformée de Fourier discrète $\{\hat{u}_k\}$
3. Introduire les corrélations en multipliant les \hat{u}_k par la densité spectrale désirée $S_0(k)$: la transformée de Fourier corrélée est donnée par $\hat{x}_k = N^{1/2} S_0(k)^{1/2} \hat{u}_k$ avec $S_0(k) = S(k) - S_{min}(k) = \sum_r N^{1/2} \exp(irk) C(r) - S_{min}(k)$ où $C(r)$ est la fonction de corrélation
4. Calculer la transformée de Fourier inverse de \hat{x}_k pour obtenir les x_j avec les corrélations souhaitées.

En particulier, nous considérons une fonction de corrélation telle que

- $C(t) \sim \exp(-t/\tau)$: corrélation décroissante exponentielle à courte portée. Cela s'apparente à un processus d'Ornstein-Uhlenbeck qui, *grosso modo*, est un processus stochastique décrivant la vitesse d'une particule Brownienne massive. Il s'agit donc d'un processus aléatoire Gaussien de moyenne nulle et de fonction de corrélation exponentielle $\phi(\tau) = \langle x(t)x(t+\tau) \rangle = \sigma^2 \exp(-\alpha|\tau|)$ avec τ le temps de latence, σ^2 la variance et α le temps de corrélation inverse du processus [48]. Une illustration des distributions de degrés ainsi que la divergence de Kullback-Leibler $D[P_{out}(k) \| P_{in}(k)]$ est reprise à la figure 3.14.
- $C(t) \sim t^{-\gamma}$: corrélation décroissante *power-law*.

Les valeurs numériques pour la divergence de Kullback-Leibler sont reprises à la table 3.1.

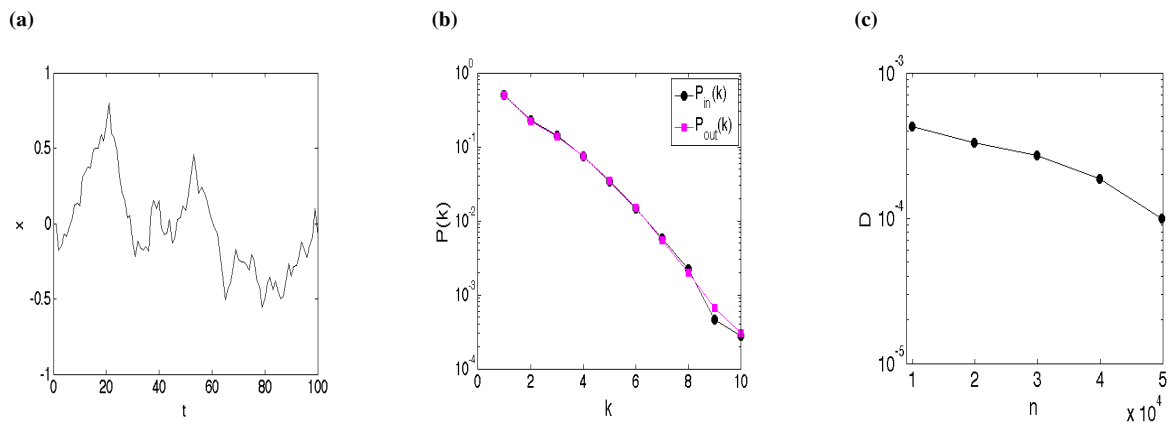


FIGURE 3.14 – (a) Série stochastique corrélée réversible (Ornstein-Uhlenbeck $\tau = 1$); (b) Distribution de degrés $P_{in}(k)$ et $P_{out}(k)$; (c) Divergence de Kullback-Leibler $D[P_{out}(k) \| P_{in}(k)]$.

■ Arnold cat map

Cette dernière est définie par les équations $x_{t+1} = x_t + y_t \pmod{1}$, $y_{t+1} = x_t + 2y_t \pmod{1}$. Nous pouvons repérer les points sur le tore \mathbb{T}^2 grâce aux deux coordonnées x et y toutes deux dans l'intervalle $[0,1]$, ce qui revient à "déplier" le tore pour en faire un carré. Il s'agit d'un exemple de système chaotique conservatif (préserve les aires). De façon intuitive, si nous raisonnons sur un carré $[0,1] \times [0,1]$ représentant le tore déplié, la "première" opération (sans le modulo) revient à étirer ce carré en un parallélogramme de largeur 2 et de hauteur 3. Le modulo va servir à ramener le tout dans le carré $[0,1] \times [0,1]$ de base (figure 3.15). Vu autrement, le déterminant vaut $\begin{vmatrix} 1 & 1 \\ 1 & 2 \end{vmatrix} = 1$ et donc la *Arnold cat map* conserve les aires.

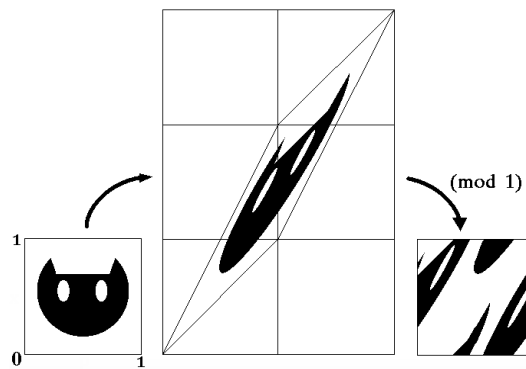


FIGURE 3.15 – Effet de la fonction chat d'Arnold sur une image carré $[0,1] \times [0,1]$. La première opération, i.e. sans le modulo étire le carré en un parallélogramme de largeur 2 et hauteur 3. La seconde opération, i.e. le modulo, sert à ramener cette image dans le carré $[0,1] \times [0,1]$ de base, [26].

L'image obtenue après une étape semble tout à fait randomisée par la transformation (figure 3.15) mais l'une des

caractéristiques de *Arnold cat map* est qu'après un nombre important de transformations, l'image revient à son état d'origine.

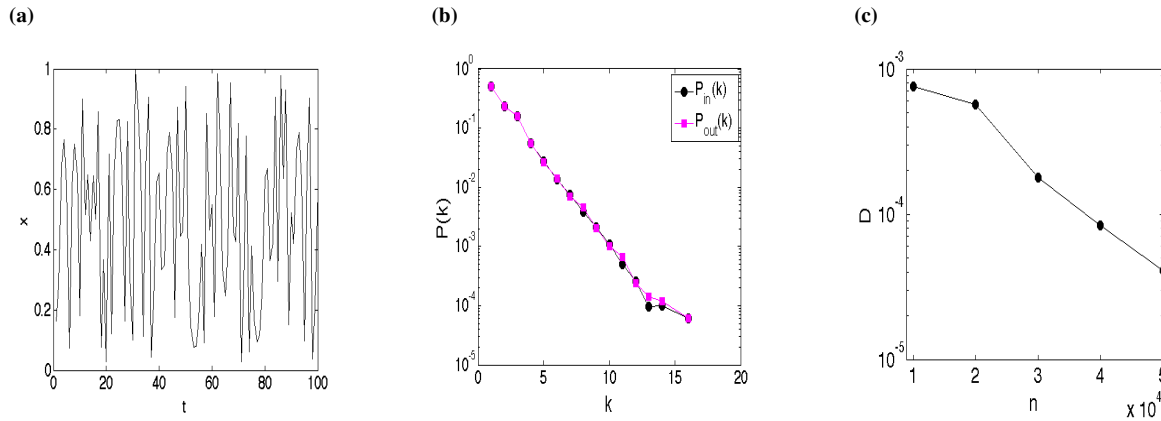


FIGURE 3.16 – (a) Série chaotique réversible (*Arnold cat map*) ; (b) Distribution de degrés $P_{in}(k)$ et $P_{out}(k)$; (c) Divergence de Kullback-Leibler $D[P_{out}(k)||P_{in}(k)]$.

Irréversibilité

Nous appliquons la méthode sur quelques systèmes chaotiques dissipatifs. De tels systèmes ne préservent pas le volume de l'espace de phase et produisent des séries temps-irréversibles. Nous montrons que dans ce cas, le degré de réversibilité associé aux différentes séries est toujours positif.

■ Logistic map $\mu = 4$

Les résultats sont repris à la figure 3.17 où nous voyons clairement sur le panel b que les distributions de degrés sont bien distinctes ce qui est corroboré par la KLD $D[P_{out}(k)||P_{in}(k)]$. Contrairement aux cas précédents, cette mesure ne converge pas vers zéro et tend même à se stabiliser à une valeur bien au-dessus de zéro.

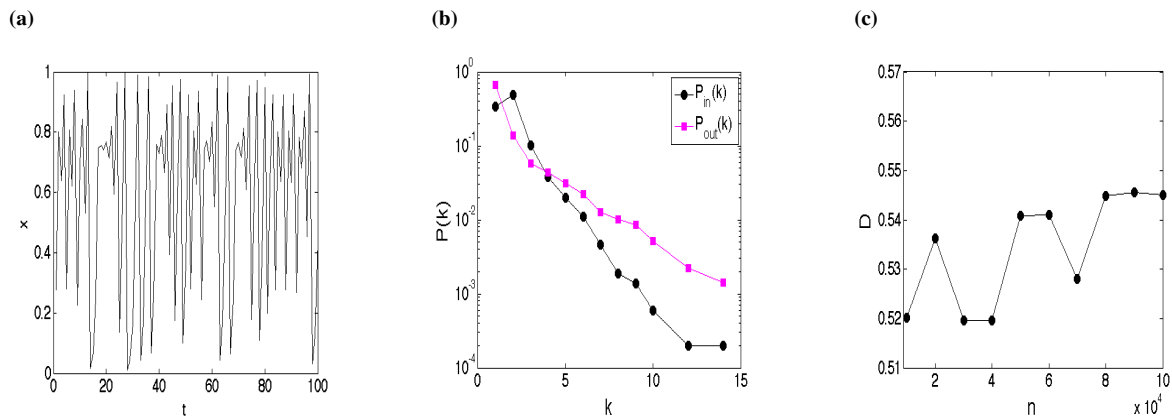


FIGURE 3.17 – (a) Série chaotique irréversible (*Logistic map* $\mu = 4$) ; (b) Distribution de degrés $P_{in}(k)$ et $P_{out}(k)$; (c) Divergence de Kullback-Leibler $D[P_{out}(k)||P_{in}(k)]$.

Nous avons montré analytiquement que pour une série stochastique non corrélée, $P_{in}(k) = P_{out}(k)$ (éq. (3.1.3)) et que par conséquent $D[P_{out}(k)||P_{in}(k)] = 0$. Dans le cas de séries stochastiques corrélées, trouver des résultats exacts pour l'entièreté des distributions de degrés se révèle être une tâche ardue [30]. Cependant, nous savons que la KLD entre deux distributions s'annule uniquement lorsque ces dernières sont égales sur tout le support. Subséquemment, il suffit de montrer que ces distributions diffèrent pour une valeur k pour que la KLD soit strictement positive et que le système chaotique soit temps-irréversible.

Soit une série temporelle $\{x_t\}_{t=1,\dots,N}$ de N données définie sur un support (a, b) avec une distribution de probabilités jointes $f(x_1, x_2, \dots, x_N)$. Soient x_{t-1} , x_t et x_{t+1} trois données de cette série. Par construction, nous avons

$$P_{out}(k=1) = P(x_t < x_{t+1}) = \int_a^b \int_{x_t}^b f(x_t, x_{t+1}) dx_{t+1} dx_t$$

$$P_{in}(k=1) = P(x_{t-1} > x_t) = \int_a^b \int_{x_t}^b f(x_{t-1}, x_t) dx_{t-1} dx_t$$
(3.17)

La probabilité d'avoir $P_{out}(k=1)$ est en fait la probabilité que la série soit strictement croissante. À l'inverse, la probabilité d'avoir $P_{in}(k=1)$ est en fait la probabilité que la série soit strictement décroissante. Notons que cette probabilité est indépendante du temps puisque nous ne considérons que des séries stationnaires. Si le modèle chaotique est de la forme $x_{t+1} = F(x_t)$, il est markovien et les équations (3.17) peuvent s'écrire

$$P_{out}(k=1) = \int_a^b \int_{x_t}^b f(x_t) f(x_{t+1}|x_t) dx_{t+1} dx_t \quad (3.18)$$

$$P_{in}(k=1) = \int_a^b \int_{x_t}^b f(x_{t-1}) f(x_t|x_{t-1}) dx_{t-1} dx_t$$

Pour les systèmes dynamiques chaotiques pour lesquels les trajectoires sont dans l'attracteur, il existe une densité de probabilité invariante qui caractérise le temps que le système passe dans les différentes régions de l'attracteur. Dans le cas du modèle logistique

$$F(x_t) = \mu x_t(1 - x_t) \quad (3.19)$$

avec le paramètre $\mu = 4$, l'attracteur est l'intervalle $[0,1]$ entier et la densité invariante $f(x)$ correspond à

$$f(x) \equiv \rho(x) = \frac{1}{\pi \sqrt{x(1-x)}}. \quad (3.20)$$

En effet, nous avons la fonction $F(x) = 4x(1-x)$. Il est facile de voir, en résolvant une équation quadratique, que $F^{-1}([0,x]) = [0, \frac{1-\sqrt{1-x}}{2}] \cup [\frac{1+\sqrt{1-x}}{2}, 1]$. Dès lors, en considérant la fonction de répartition de X_{n+1} , nous avons

$$\begin{aligned} \mathbb{P}(X_{n+1} < x) &= \mathbb{P}(X_{n+1} \in [0, x]) \\ &= \mathbb{P}(X_n \in F^{-1}([0, x])) \\ &= \mathbb{P}(X_n \in \left([0, \frac{1-\sqrt{1-x}}{2}] \cup [\frac{1+\sqrt{1-x}}{2}, 1] \right)) \\ &= \int_0^{\frac{1-\sqrt{1-x}}{2}} \rho_n(y) dy + \int_{\frac{1+\sqrt{1-x}}{2}}^1 \rho_n(y) dy \end{aligned}$$

où ρ_n est la densité de X_n . Dès lors, nous obtenons

$$\int_0^x \rho_{n+1}(y) dy = \int_0^{\frac{1-\sqrt{1-x}}{2}} \rho_n(y) dy + \int_{\frac{1+\sqrt{1-x}}{2}}^1 \rho_n(y) dy.$$

Pour résoudre cette équation, nous dérivons les deux côtés de l'égalité par rapport à x . Le côté gauche donne directement la densité de $\rho_{n+1}(x)$, ce que nous cherchons.

$$\begin{aligned} \rho_{n+1}(x) &= \frac{d}{dx} \int_0^{\frac{1-\sqrt{1-x}}{2}} \rho_n(y) dy + \frac{d}{dx} \int_{\frac{1+\sqrt{1-x}}{2}}^1 \rho_n(y) dy \\ &= \rho_n\left(\frac{1-\sqrt{1-x}}{2}\right) \frac{d}{dx} \left(\frac{1-\sqrt{1-x}}{2}\right) + \rho_n\left(\frac{1+\sqrt{1-x}}{2}\right) \frac{d}{dx} \left(\frac{1+\sqrt{1-x}}{2}\right) \\ &= \frac{1}{4\sqrt{1-x}} \left[\rho_n\left(\frac{1-\sqrt{1-x}}{2}\right) + \rho_n\left(\frac{1+\sqrt{1-x}}{2}\right) \right]. \end{aligned} \quad (3.21)$$

Remarquons que cela définit un opérateur, notons le T . Le fait d'avoir une mesure de probabilité invariante ρ se traduit par $\rho = T\rho$. Il ne reste qu'à montrer que l'éq. (3.20) respecte cette égalité. en outre,

$$\begin{aligned} \rho\left(\frac{1-\sqrt{1-x}}{2}\right) &= \frac{1}{\pi} \left[\frac{1-\sqrt{1-x}}{2} \left(1 - \frac{1+\sqrt{1-x}}{2}\right) \right]^{-1/2} \\ &= \frac{2}{\pi \sqrt{x}} \\ &= \rho\left(\frac{1+\sqrt{1-x}}{2}\right) \end{aligned} \quad (3.22)$$

Enfin, en utilisant l'éq. (3.22), nous avons

$$\begin{aligned} T\rho &= \frac{1}{4\sqrt{1-x}} \left(\frac{2}{\pi \sqrt{x}} + \frac{2}{\pi \sqrt{x}} \right) \\ &= \frac{1}{\pi \sqrt{x(1-x)}} \\ &= \rho, \end{aligned} \quad (3.23)$$

ce qui montre bien l'éq. (3.20).

Maintenant, pour un système déterministe, la probabilité de transition est donnée simplement par

$$f(x_{t+1}|x_t) = \delta(x_{t+1} - F(x_t)),$$

avec δ la distribution delta de Dirac. Dès lors, pour le modèle logistique avec $\mu = 4$ et $x \in [0, 1]$, nous pouvons réécrire les équations (3.18) comme

$$\begin{aligned} P_{out}(k=1) &= \int_0^1 \int_{x_t}^1 f(x_t) \delta(x_{t+1} - F(x_t)) dx_{t+1} dx_t \\ P_{in}(k=1) &= \int_0^1 \int_{x_t}^1 f(x_{t-1}) \delta(x_t - F(x_{t-1})) dx_{t-1} dx_t \end{aligned} \quad (3.24)$$

Nous avons, par les propriétés de la distribution de Dirac, que l'intégrale $\int_{x_t}^1 \delta(x_{t+1} - F(x_t)) dx_{t+1}$ vaut 1 uniquement lorsque $F(x_t)$ appartient au support i.e., lorsque $F(x_t) \in [x_t, 1]$. Par la définition de $F(x_t)$, cela est le cas lorsque $x_t \in]0, 3/4[$. Dans les autres cas, l'intégrale égale 0. Cela a pour effet de restreindre le support de l'intégrale (3.24). Dès lors, en remplaçant cela dans l'équation (3.24), nous obtenons

$$P_{out}(k=1) = \int_0^{3/4} f(x_t) dx_t = \frac{2}{3}. \quad (3.25)$$

Maintenant, pour la distribution de degrés entrants, nous avons

$$\int_{x_t}^1 f(x_{t-1}) \delta(x_t - F(x_{t-1})) dx_{t-1} = \sum_{x^* | F(x^*) = x_t} f(x^*) / |F'(x^*)|, \quad (3.26)$$

c'est-à-dire que nous sommions sur toutes les racines x^* de l'équation $F(x) = x_t$, si et seulement si $F(x_{t-1}) > x_t$. Mais puisque $x_{t-1} \in [x_t, 1]$ dans l'intégrale précédente, il est facile de voir que cela est vérifié si et seulement si $x_t \in]0, 3/4[$. En calculant les racines de $F(x) = x_t$, nous trouvons une solution $x^* = (1 + \sqrt{1 - x_t})/2$. En utilisant cela dans l'équation (3.26) et en réarrangeant avec l'équation (3.24), nous obtenons finalement

$$P_{in}(k=1) = \int_0^{3/4} \frac{f(x^*)}{4\sqrt{1-x_t}} dx_t = \frac{1}{3}. \quad (3.27)$$

Nous venons donc de montrer que $P_{in}(k=1) \neq P_{out}(k=1)$ pour le modèle logistique et par conséquent, la KLD basée sur la distribution des degrés est non nulle et donc positive comme le montre la figure 3.17.

- α map : $x_{t+1} = 1 - |2x_t - 1|^\alpha$ avec $\alpha = 3$ et 4
- Henon map : $x_{t+1} = 1 + y_t - ax_t^2$, $y_{t+1} = bx_t$ avec $a = 1.4$ et $b = 0.3$

Les différents résultats numériques sont repris à la table 3.1.

Séries	$D[P_{out}(k) P_{in}(k)]$
$U([0,1])$ non corrélée	3.56e-4
Ornstein-Uhlenbeck $\tau = 1$	1.02e-4
Long-range (power law) correlated stationary process ($\gamma = 2$)	3.01e-5
Logistic map ($\mu = 4$)	0.545
α map ($\alpha = 3$)	0.653
α map ($\alpha = 4$)	0.788
Henon map	0.264
Arnold cat map	4.12e-5

TABLE 3.1 – Valeurs des mesures d'irréversibilité associées à la divergence de Kullback-Leibler $D[P_{out}(k)||P_{in}(k)]$ pour différentes séries de 50 000 données. KLD $D[P_{out}(k)||P_{in}(k)]$ parvient à distinguer les processus réversibles des processus irréversibles en s'annulant pour les premiers et en prenant une valeur positives pour les seconds.

3.3 Conclusion

Nous avons présenté une méthode permettant l'analyse des séries temporelles à travers des réseaux. Les graphes de visibilité naturelle, horizontale (dirigé) ont été définis ainsi que quelques propriétés. Cette méthode est très simple d'utilisation et convient également pour des séries temporelles de grandes tailles. Bien qu'il ne s'agisse que d'une faible

amorce, nous avons exposé quelques résultats qui montrent comment certaines propriétés des signaux sont transférées dans les graphes associés. En particulier, l'estimation de l'exposant de Hurst, le problème du filtrage de bruit ou encore déterminer si une série temporelle est temps-réversible ou non. Pour cela, nous avons uniquement utilisé les distributions de degrés.

Chapitre 4

Taux d'information mutuelle

Dans ce chapitre, nous nous concentrons sur une méthode basée sur l'information pour déduire la connectivité des systèmes complexes à partir de données d'une série temporelle. Dans un système, de l'information peut être produite et transférée entre les différents composants du système. Lorsqu'il y a transfert, cela signifie qu'il y a au moins deux composants qui interagissent entre eux, de façon directe ou indirecte. Nous étudions ici la quantité d'information transférée par unité de temps (MIR) entre deux composantes (séries temporelles) en vue de déterminer si un lien existe entre elles. La référence principale pour ce chapitre est [8].

4.1 Aspects théoriques

Nous introduisons ici les quelques notions de théories nécessaires pour le calcul du MIR.

4.1.1 Entropie de Shannon

L'entropie de Shannon est un concept mathématique correspondant à la quantité d'information contenue ou délivrée par une source d'information [54]. Plus la source émet d'informations différentes, plus l'entropie, c'est-à-dire l'incertitude sur ce que la source émet, est grande. De cette façon, si une source est réputée pour envoyer toujours la même information, par exemple le symbole 'a', son entropie est nulle puisque nous sommes certains que le prochain symbole sera un 'a'. À l'inverse, si la source est réputée envoyer un 'a' la moitié du temps et un 'b' l'autre moitié, le récepteur ne connaît pas avec certitude la prochaine lettre tirée et par conséquent, l'entropie n'est pas nulle. L'entropie quantifie l'information nécessaire pour pouvoir déterminer sans ambiguïté ce que la source transmet. Au plus le récepteur connaît d'informations sur ce que la source envoie, au plus l'entropie est petite. Lorsque que tous les symboles sont équiprobables, l'entropie est maximale.

Dans le cas où nous disposons de $N = 2^n$ symboles équiprobables, il suffit de poser n questions pour connaître le symbole envoyé par la source. Dans ce cas, la quantité d'information contenue par le symbole vaut $n = \log_2 N$. Remarquons que nous gardons cette formule même dans le cas où N n'est pas une puissance de 2.

Supposons maintenant que nous puissions répartir nos N symboles en n sous-catégories. Chaque catégorie i contient N_i symboles ($N = \sum_{i=1}^n N_i$). Soit X la variable aléatoire donnant la catégorie du symbole considéré. La probabilité que le symbole considéré appartienne à la catégorie i est définie par $P_i = N_i/N$. Pour pouvoir déterminer le symbole en question, nous procédons en deux temps. Tout d'abord, nous devons identifier sa catégorie i , ce qui requiert une quantité d'information $H(X)$. Ensuite, nous déterminons le symbole au sein de la catégorie i , exigeant une quantité d'information $\log_2(N_i)$. Étant donné que les chances que le symbole considéré appartienne à la catégorie i s'élèvent à P_i , la quantité moyenne d'information nécessaire pour déterminer le symbole connaissant sa catégorie est

$$\sum_{i=1}^n P_i \log_2(N_i). \quad (4.1)$$

Dès lors, pour connaître la quantité d'information totale $\log_2(N)$, nous devons y ajouter la quantité $H(X)$ qui détermine sa catégorie

$$\log_2(N) = H(X) + \sum_{i=1}^n P_i \log_2(N_i). \quad (4.2)$$

Ce qui peut se réécrire comme

$$H(X) = \log_2(N) - \sum_{i=1}^n P_i \log_2(N_i) = - \sum_{i=1}^n P_i \log_2\left(\frac{N_i}{N}\right) = - \sum_{i=1}^n P_i \log_2(P_i) \quad (4.3)$$

En gros, l'entropie H de la source X comprenant n symboles $\{x_1, x_2, \dots, x_n\}$ avec probabilité d'apparition p_1, p_2, \dots, p_n respectivement est définie comme suit

$$H(X) = -E(\log_2 P(X = x_i)) = \sum_{i=1}^n P_i \log_2 \left(\frac{1}{P_i} \right) = - \sum_{i=1}^n P_i \log_2 P_i \quad (4.4)$$

où E désigne l'espérance mathématique. Dans le cas de deux variables aléatoires X et Y , nous définissons l'entropie conjointe

$$H(X, Y) = - \sum_{i,j=1}^n P(X = x_i, Y = y_j) \log_2 P(X = x_i, Y = y_j) \quad (4.5)$$

ainsi que l'entropie conditionnelle de Y par rapport à X

$$H(X|Y) = - \sum_{i,j=1}^n P(X = x_i|Y = y_j) \log_2 P(X = x_i|Y = y_j). \quad (4.6)$$

Quelques propriétés

- $0 \leq H(X) \leq \log_2(n)$. En effet, par définition, l'entropie est toujours positive étant donné que P_i est une probabilité donc comprise entre 0 et 1. Lorsque $P_i = 0$, nous posons $H(X) = 0$. De plus, l'entropie est maximale lorsque tous les états ont la même probabilité d'apparition ; dans ce cas, $P_i = \frac{1}{n}$. Dès lors $H(X) = - \sum_i \frac{1}{n} \log_2 \left(\frac{1}{n} \right) = \log_2(n)$;
- l'entropie est symétrique : $H(X, Y) = H(Y, X)$;
- $H(X, Y) \leq H(X) + H(Y)$ avec égalité lorsque les variables aléatoires sont indépendantes puisque dans ce cas, $P(X = x, Y = y) = P(X = x)P(Y = y)$. Dès lors,

$$\begin{aligned} H(X, Y) &= - \sum_{x \in X} \sum_{y \in Y} P(x, y) \log P(x, y) \\ &= - \sum_{x \in X} \sum_{y \in Y} P(x)P(y) \log(P(x)P(y)) \\ &= - \sum_{x \in X} \sum_{y \in Y} P(x)P(y) (\log P(x) + \log P(y)) \\ &= - \sum_{x \in X} \sum_{y \in Y} P(x)P(y) \log P(x) - \sum_{x \in X} \sum_{y \in Y} P(x)P(y) \log P(y) \\ &= - \sum_{x \in X} P(x) \log P(x) \sum_{y \in Y} P(y) - \sum_{y \in Y} P(y) \log P(y) \sum_{x \in X} P(x) \\ &= - \sum_{x \in X} P(x) \log P(x) - \sum_{y \in Y} P(y) \log P(y) \\ &= H(X) + H(Y) ; \end{aligned}$$

- $H(X, Y) = H(X) + H(X|Y)$.

4.1.2 Information mutuelle

L'information mutuelle (MI) entre deux variables aléatoires X et Y d'un système est la quantité d'incertitude que nous avons pour X (ou Y) après avoir observé Y (ou X). Formellement, cela s'écrit

$$MI_{XY}(N) = H(X) + H(Y) - H(X, Y) \quad (4.7)$$

où

- N est le nombre d'éléments des variables aléatoires X et Y ;
- $H(X) = - \sum_{i=1}^N P_X(i) \log(P_X(i))$, l'entropie marginale de X ;
- $H(Y) = - \sum_{j=1}^N P_Y(j) \log(P_Y(j))$, l'entropie marginale de Y ;
- $H(X, Y) = - \sum_{i,j=1}^{N^2} P_{XY}(i, j) \log(P_{XY}(i, j))$, l'entropie conjointe entre X et Y .

$P_X(i)$ représente la probabilité qu'un événement i se produise dans X , $P_Y(j)$ représente la probabilité qu'un événement j se produise dans Y et $P_{XY}(i, j)$ représente la probabilité qu'un événement i se produise dans X simultanément à un événement j dans Y . Dès lors, par les propriétés du logarithme, nous pouvons exprimer l'éq. (4.7) comme

$$MI_{XY}(N) = \sum_{i=1}^N \sum_{j=1}^N P_{XY}(i, j) \log \left(\frac{P_{XY}(i, j)}{P_X(i)P_Y(j)} \right) \quad (4.8)$$

qui peut être interprété comme la force d'indépendance entre les variables X et Y . Lorsque ces dernières sont indépendantes, $MI_{XY} = 0$.

Il existe plusieurs façons de calculer MI_{XY} . La première est la méthode des histogrammes qui consiste à trouver une partition de l'espace en deux dimensions en partitionnant de façon égale ou bien de façon "adaptive" [13, 18]. La seconde méthode fait usage de noyaux de densité, où le noyau de la fonction de densité de probabilité est estimé [37]. Enfin, la dernière méthode calcule MI_{XY} par estimation de probabilités à partir des distances entre les plus proches voisins [25]. Nous retiendrons la première méthode. Pour ce faire, nous considérons l'espace probabilisable Ω généré par X et Y . Ensuite, nous quadrillons cet espace Ω en $N \times N$ cellules dont le côté mesure $\varepsilon = 1/N$. Dès lors, la probabilité $P_X(i)$ qu'un événement i se produise dans X est le rapport entre le nombre de points situés dans la ligne i et le nombre de points total. De façon analogue, la probabilité $P_Y(j)$ qu'un événement j se produise dans Y est la fraction de points situés dans la colonne j de Ω . Finalement, $P_{XY}(i, j)$ est la fraction de points dans la case (i, j) de la partition Ω , $i, j = 1, \dots, N$. Il est dès lors évident que la quantité MI_{XY} dépende de la partition considérée. Malheureusement, dans le cas des systèmes aléatoires ainsi que pour des partitions non-Markov, cette approche surestime la valeur de MI_{XY} et cela pour deux raisons : la résolution finie d'une partition non-Markov et la longueur finie des séries temporelles [22]. Il semblerait donc que ces erreurs soient inévitables puisque nous travaillons toujours avec des séries finies. Pour contrer ce petit désagrément, nous introduisons une normalisation basée sur le taux d'information mutuelle (MIR).

4.1.3 Taux d'information mutuelle

Comme mentionné précédemment, MI_{XY} a le mauvais goût de dépendre de la partition considérée. Pour passer outre ce problème, nous définissons le taux d'information mutuelle qui est invariant aux changements de partitions. Pour des séries temporelles de taille infinie, le MIR se définit comme étant l'information mutuelle échangée par unité de temps entre X et Y . En dénotant par $MI_{XY}(L, N)$ le MI défini en (4.7) entre les variables aléatoires X et Y sur des trajectoires de longueur L à travers un quadrillage composé de N cellules, nous pouvons exprimer le MIR_{XY} comme

$$\begin{aligned} MIR_{XY} &= \lim_{N \rightarrow \infty} \lim_{L \rightarrow \infty} \sum_{i=1}^{L-1} \frac{MI_{XY}(i+1, N) - MI_{XY}(i, N)}{L} \\ &= \lim_{N \rightarrow \infty} \lim_{L \rightarrow \infty} \frac{MI_{XY}(L, N) - MI_{XY}(1, N)}{L} \\ &= \lim_{N \rightarrow \infty} \lim_{L \rightarrow \infty} \frac{MI_{XY}(L, N)}{L} \end{aligned} \quad (4.9)$$

puisque $\lim_{L \rightarrow \infty} \frac{MI_{XY}(1, N)}{L} = 0$.

Lorsque la partition composée de N cellules est une partition de Markov d'ordre T , le MIR peut s'exprimer de la façon suivante

$$MIR_{XY} = \frac{MI_{XY}(N)}{T(N)} \quad (4.10)$$

avec N et $T(N)$ des quantités finies. Notons qu'il doit y avoir suffisamment de points dans chaque cellule pour générer des probabilités significatives.

Étant donné que la partition est d'ordre T , les points contenus dans une cellule se propagent à travers tout l'ensemble après T itérations. $T(N)$ représente le temps le plus court pour que la corrélation entre X et Y tombe à zéro pour la partition de Markov considérée. $T(N)$ représente également le temps le plus court après lequel l'évolution du système devient non prédictible. Pour calculer cette valeur $T(N)$, nous définissons $T(N)$ comme étant le diamètre d'un graphe G , basé sur la dynamique des points envoyés d'une cellule de Ω à une autre c'est-à-dire le graphe dont la connectivité est donnée par les transitions de points d'une cellule à une autre de Ω . Pour construire ce graphe, nous fonctionnons de la manière suivante. Chaque cellule de Ω contenant au moins un point est associée à un nœud dans G . Lorsqu'un point passe d'une cellule i à une cellule j , nous créons un lien entre le nœud i et le nœud j de G . Lorsque le lien existe, nous lui donnons un poids de 1 et 0 dans le cas contraire. Par conséquent, la matrice G est une matrice binaire. En ayant construit G de cette façon, nous comprenons mieux pourquoi nous avons défini $T(N)$ comme étant son diamètre. En effet, $T(N)$ est le temps minimal pour qu'un point à l'intérieur d'une cellule se propage à travers tout l'espace Ω tandis que le diamètre d'un graphe est défini comme étant la distance minimale requise pour traverser ce graphe.

En vue d'estimer le MIR à partir des données d'une série temporelle de longueur finie, nous tronquons la série de l'expression (4.9) à une taille finie N et considérons également des morceaux de trajectoires de longueur L de la série. La taille N et la longueur L dépendent respectivement de la résolution des données et de la longueur totale de la série temporelle

$$MIR_{XY} \cong \frac{1}{L} \sum_{i=1}^L (MI_{XY}(i+1, N) - MI_{XY}(i, N)) \quad (4.11)$$

où l'égalité tient lorsque la partition est de Markov. La longueur L représente aussi l'ordre maximal T d'une partition générant des probabilités statistiquement significatives, construite à partir de ces morceaux de trajectoire. Si nous supposons que $T = L$, nous obtenons

$$MIR_{XY} \cong \frac{1}{T} \sum_{i=1}^T (MI_{XY}(i+1, N) - MI_{XY}(i, N)). \quad (4.12)$$

Considérons à présent deux partitions, Λ_1 et Λ_2 avec des corrélations temporelles de décroissance de T_1 et T_2 respectivement et dont le nombre de cellules sont $N_1 \times N_1$ et $N_2 \times N_2$ respectivement, avec $N_1 < N_2$ et $T_2 = T_1 + 1$. De plus, Λ_1 et Λ_2 sont reliés par l'opérateur d'évolution F de la façon suivante : $F^{-1}(\Lambda_1) = \Lambda_2$, c'est-à-dire que Λ_2 est la pré-itération de Λ_1 . Dès lors,

$$\begin{aligned} \text{MI}_{XY}(T_2, \Lambda_1) &= \text{MI}_{XY}(T_1 + 1, F(\Lambda_2)) \\ &= \text{MI}_{XY}(T_1, \Lambda_2) \end{aligned} \quad (4.13)$$

Par conséquent, nous pouvons réécrire (4.12) comme

$$\begin{aligned} \text{MIR}_{XY} &\cong \frac{1}{T_1} \sum_{i=1}^{T_1} (\text{MI}_{XY}(i+1, \Lambda_1) - \text{MI}_{XY}(i, \Lambda_1)) \\ &\cong \frac{1}{T_1} \sum_{i=1}^{T_1} (\text{MI}_{XY}(i, \Lambda_2) - \text{MI}_{XY}(i, \Lambda_1)). \end{aligned} \quad (4.14)$$

Dans le cas de partitions de Markov, nous avons

$$\begin{aligned} \text{MI}_{XY}(i, \Lambda_k) &= \text{MI}_{XY}(i-1, \Lambda_{k+1}) \\ &= \text{MI}_{XY}(i-2, \Lambda_{k+2}) \\ &\dots \\ &= \text{MI}_{XY}(1, \Lambda_{k+i-1}) \end{aligned} \quad (4.15)$$

Dans le cas de partitions non Markov mais proches, nous pouvons approximer

$$\begin{aligned} \text{MIR}_{XY} &\cong \frac{1}{T_1} \sum_{i=1}^{T_1} (\text{MI}_{XY}(i+1, \Lambda_1) - \text{MI}_{XY}(i, \Lambda_1)) \\ &= \frac{1}{T_1} (\text{MI}_{XY}(T_1 + 1, \Lambda_1) - \text{MI}_{XY}(1, \Lambda_1)) \\ &\simeq \frac{1}{T_1} (\text{MI}_{XY}(1, \Lambda_{T_1+1}) - \text{MI}_{XY}(1, \Lambda_1)) \\ &\equiv \frac{1}{T_1} \text{MI}_{XY}(1, \Lambda_{T_1}) \end{aligned} \quad (4.16)$$

et nous retombons bien sur l'expression (4.10).

Pour qu'une partition atteigne le statut de "proche de Markov", il suffit, par exemple, que la partition soit construite à partir d'un nombre suffisamment grand de cellules de tailles identiques de longueur $\varepsilon = 1/N$. Étant donné que nous considérons justement ce type de partitions, nous pouvons donc bien utiliser l'approximation.

Notons également que le MIR_{XY} dépend de la partition considérée. Cela n'est pas le cas seulement lorsque la partition est de Markov. Le calcul de MI_{XY} se fait à partir de (4.7), ce qui exige le calcul des probabilités de Ω . Lorsque le nombre moyen de points à l'intérieur d'une cellule de la partition Ω , $\langle N_0(N) \rangle$, est suffisamment élevé

$$\langle N_0(N) \rangle \geq N_{oc}, \quad (4.17)$$

les probabilités calculées sont considérées non biaisées.

Cependant, le MIR pour des partitions non Markov possède non seulement une dépendance vis-à-vis du nombre de cellules de la partition mais présente également une erreur systématique. Puisque le MIR_{XY} d'une partition non Markov de $N \times N$ cellules de même taille est dépendant de la partition, nous définissons une autre mesure, basée sur le MIR_{XY} , qui est cette fois-ci, indépendante de la partition et qui est appropriée pour l'inférence de réseau.

4.1.4 Inférence de réseau avec MIR

Supposons un réseau composé de M nœuds. Pour déduire la structure de ce réseau, nous calculons le MIR pour chacune des $M(M-1)/2$ différentes paires de nœuds du graphes. En effet, étant donné que le MIR est symétrique par définition, $\text{MIR}_{XY} = \text{MIR}_{YX}$, et puisque nous ne sommes intéressés que par l'échange d'information entre différentes variables, c'est-à-dire que nous ne nous occupons pas de MIR_{XX} , il ne nous reste que $M(M-1)/2$. Le MIR_{XY} entre chaque nœud nous est donné par la valeur moyenne calculée sur différentes tailles de partition N_i : $\text{MIR}_{XY} = E_i(\text{MIR}_{XY}(N_i))$. En vue d'éliminer l'erreur dans le calcul de cette quantité, nous effectuons en réalité une moyenne pondérée. Étant donné que plus une partition est fine, c'est-à-dire avec N grand, plus elle est proche d'être de Markov, il semble dès lors évident qu'une partition plus fine contribue plus au MIR_{XY} qu'une partition plus grossière dans la moyenne pondérée.

Dès lors, nous introduisons une nouvelle normalisation pour le MIR. Nous utilisons une grille pour laquelle les $N \times N$ cellules ont la même taille. Ayant calculé le $\text{MIR}_{XY}(N)$ pour chaque paire de nœuds, nous lui soustrayons sa valeur minimale et dénotons la quantité ainsi obtenue par $\min\{\text{MIR}_{XY}(N)\}$. En théorie, une paire qui est déconnectée devrait avoir une valeur de MIR proche de zéro mais en pratique, à cause des erreurs systématiques dues à l'utilisation de partition non Markov, cela n'est pas le cas. Nous effectuons donc cette soustraction en vue de réduire cette erreur (surestimation) du MIR. Ensuite, étant donné que le résultat de la soustraction dépend de la taille N , nous normalisons cette quantité par $\max\{\text{MIR}_{XY}(N)\} - \min\{\text{MIR}_{XY}(N)\}$, où le maximum et le minimum sont pris parmi toutes les paires. Dès lors, nous obtenons

$$\hat{\text{MIR}}_{XY}(N) = \frac{\text{MIR}_{XY}(N) - \min\{\text{MIR}_{XY}(N)\}}{\max\{\text{MIR}_{XY}(N)\} - \min\{\text{MIR}_{XY}(N)\}} \quad (4.18)$$

où

- $\text{MIR}_{XY}(N)$ est le MIR entre les nœuds X et Y ;
- $\min\{\text{MIR}_{XY}(N)\}$ est le minimum parmi les $M(M-1)/2$ paires ;
- $\max\{\text{MIR}_{XY}(N)\}$ est le maximum parmi les $M(M-1)/2$ paires.

Bien que nous n'ayons plus d'erreur systématique dans le calcul, $\hat{\text{MIR}}_{XY}(N)$ est toujours fonction de N . Pour ne plus avoir cette dépendance, nous effectuons une moyenne sur différentes valeurs de N . En particulier, nous appliquons la formule (4.18) pour différentes tailles de réseaux N_i , $i = 1, \dots, m$ où N_m est la plus petite taille. Nous obtenons donc $\overline{\text{MIR}}_{XY}(N_i)$. Encore une fois, nous effectuons une normalisation

$$\overline{\text{MIR}}_{XY} = \frac{\sum_i \hat{\text{MIR}}_{XY}(N_i)}{\max\{\sum_i \hat{\text{MIR}}_{XY}(N_i)\}} \quad (4.19)$$

où le maximum est pris parmi les N_m grilles.

Finalement, en appliquant l'expression (4.19) à chaque paire XY , nous obtenons une valeur moyenne dénotée par $\overline{\text{MIR}}_{XY}$. Une valeur élevée du $\overline{\text{MIR}}_{XY}$ signifie qu'une grande quantité d'information par unité de temps est échangée entre X et Y .

Afin de réaliser l'inférence du réseau à partir du MIR, nous fixons un seuil $\sigma \in [0, 1]$ et construisons une matrice d'adjacence A^c . A_{XY}^c prend la valeur 1 lorsque le $\overline{\text{MIR}}_{XY}$ est supérieur au seuil fixé et vaut 0 dans le cas contraire. Dans les exemples qui suivent, nous connaissons la matrice initiale A . Cela nous permet de comparer les deux matrices A^c et A et de déterminer le pourcentage de bonne inférence. Dans les cas pratiques, nous ne connaissons pas la topologie à inférer et ne pouvons donc calculer la différence entre les matrices A et A^c et il est plus délicat d'affirmer si nous avons bien inféré la structure du réseau, comme expliqué dans la section suivante.

4.2 Quelques résultats

En vue d'appliquer cette méthode, nous choisissons différentes topologies de réseaux et différentes dynamiques pour les composantes des systèmes considérés. Ici, l'inférence la structure topologique d'un réseau est faite à partir des séries temporelles enregistrées sur chaque nœud.

4.2.1 Modèles

Pour nos applications, nous utilisons la dynamique décrite par l'équation suivante

$$x_{n+1}^i = f(x_n^i, r)(1 - \alpha) + \frac{\alpha}{k_i} \sum_{j=1}^M A_{ij} f(x_n^j, r)(1 - \alpha) \quad (4.20)$$

où x_n^i est l'itéré n du nœud i , $i = 1, \dots, M$ avec M le nombre de nœuds du système. $\alpha \in [0, 1]$ est la force de couplage, A est la matrice d'adjacence (binaire) qui définit le connectivité du réseau, r est le paramètre dynamique de chaque fonction, $k_i = \sum_{j=1}^M A_{ij}$ est le degré du nœuds i et $f(x_n, r)$ est la fonction considérée. Plus particulièrement, nous considérons la fonction logistique

$$f(x_n, r) = rx_n(1 - x_n) \quad (4.21)$$

avec $r = 4$ qui correspond au chaos, ainsi que la fonction *circle*

$$f(x_n, r) = x_n + r - \frac{K}{2\pi} \sin(2\pi x_n) \bmod 1 \quad (4.22)$$

avec $r = 0.35$ et $K = 6.9115$.

Sur la figure 4.1, nous avons les différentes topologies de réseaux décrites par la matrice d'adjacence A dans nos exemples. Ces réseaux sont utilisés pour étudier la robustesse de la méthode en fonction de la force de couplage, de la longueur des séries temporelles et du bruit.

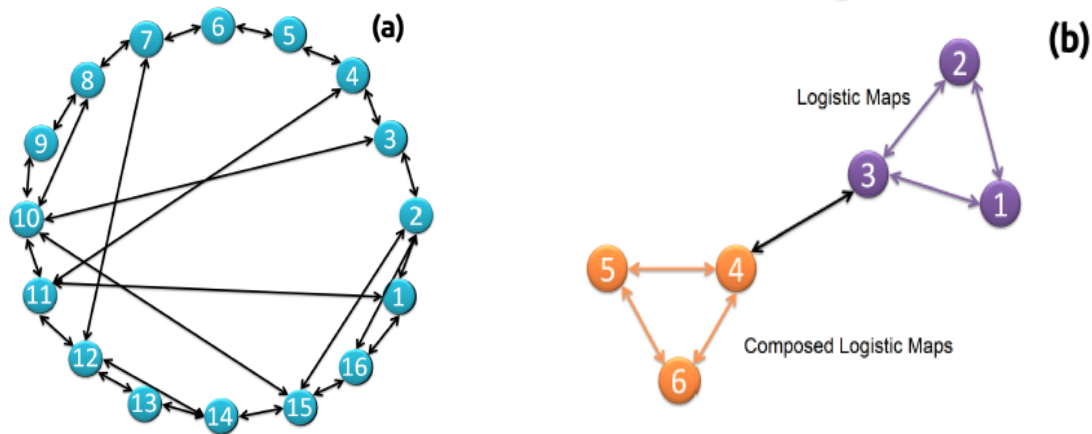


FIGURE 4.1 – Graphes utilisés pour la construction des systèmes. Le graphe (a) contient 16 nœuds dont la dynamique est soit la fonction logistique, soit la fonction *circle*. Le graphe (b) est composé de 6 nœuds formant 2 clusters dont les dynamiques diffèrent [8].

4.2.2 Forces de couplage

Pour cette sous-section, nous utilisons la structure du graphe donné à la figure 4.1a. La dynamique de chaque nœud est décrite soit par la fonction logistique, soit par la fonction *circle*. Nous analysons l'efficacité de la méthode lorsque la force de couplage α varie. Il est possible de montrer que pour les logistic et circle maps, la dynamique est quasi-périodique lorsque $\alpha > 0.15$ [47]. Dès lors, nous choisissons deux valeurs $\alpha = 0.05$ et $\alpha = 0.12$ correspondant tous deux à une dynamique chaotique. Les résultats de l'inférence du réseau sont repris à la figure 4.2. La bande jaune correspond aux seuils pour lesquels nous atteignons 100% d'inférence correcte c'est-à-dire que seuls les liens présents dans le graphe de base sont détectés. Au plus cette bande est large, au plus la probabilité de reconstruire le "bon" graphe est grande et par conséquent, au plus la méthode est robuste. Une erreur de reconstruction peut venir du fait qu'on infère un lien inexistant (erreur positive) ou au contraire, qu'on manque un lien (erreur négative). Lorsqu'une erreur se produit, nous diminuons le pourcentage d'une quantité $e\% = 100 \cdot \frac{1}{N(N-1)}$. La figure 4.3 montre qu'au plus la force de couplage α est grand (proche de 1), au plus les orbites sont synchrones.

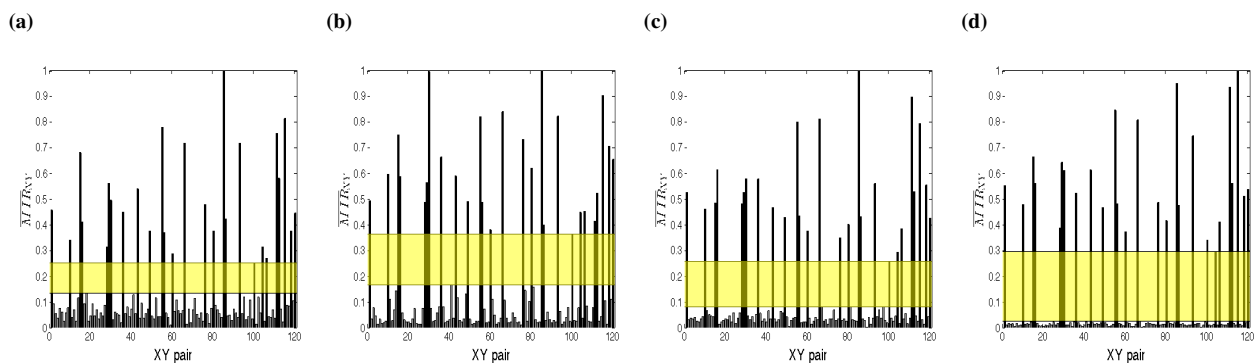


FIGURE 4.2 – Inférence de réseau pour différentes forces de couplage et différentes fonctions. (a) et (b) représentent les valeurs du MIR_{XY} des différents couples d'un réseau où la dynamique est la logistic map avec $\alpha = 0.05$ et 0.12 respectivement tandis que le fonction *circle* est utilisée pour (c) et (d). La bande jaune représente les seuils pour lesquels nous parvenons à 100% d'inférence correcte.

Lorsque nous ne connaissons pas la structure du réseau, nous ne pouvons pas calculer les différences entre la topologie réelle et celle inférée. Dans ce cas, nous pouvons trouver des seuils candidats en considérant les discontinuités dans les valeurs ordonnées du MIR. Pour la figure 4.4, nous avons considéré le graphe de la figure 4.1a avec de séries temporelles de 30000 données ainsi que différentes forces de couplage : $\alpha = 0, 0.07, 0.7$. Nous remarquons que pour $\alpha = 0$, il n'y a pas de discontinuité (figure 4.4d) puisqu'il s'agit là d'une situation "sans couplage"; les valeurs du MIR

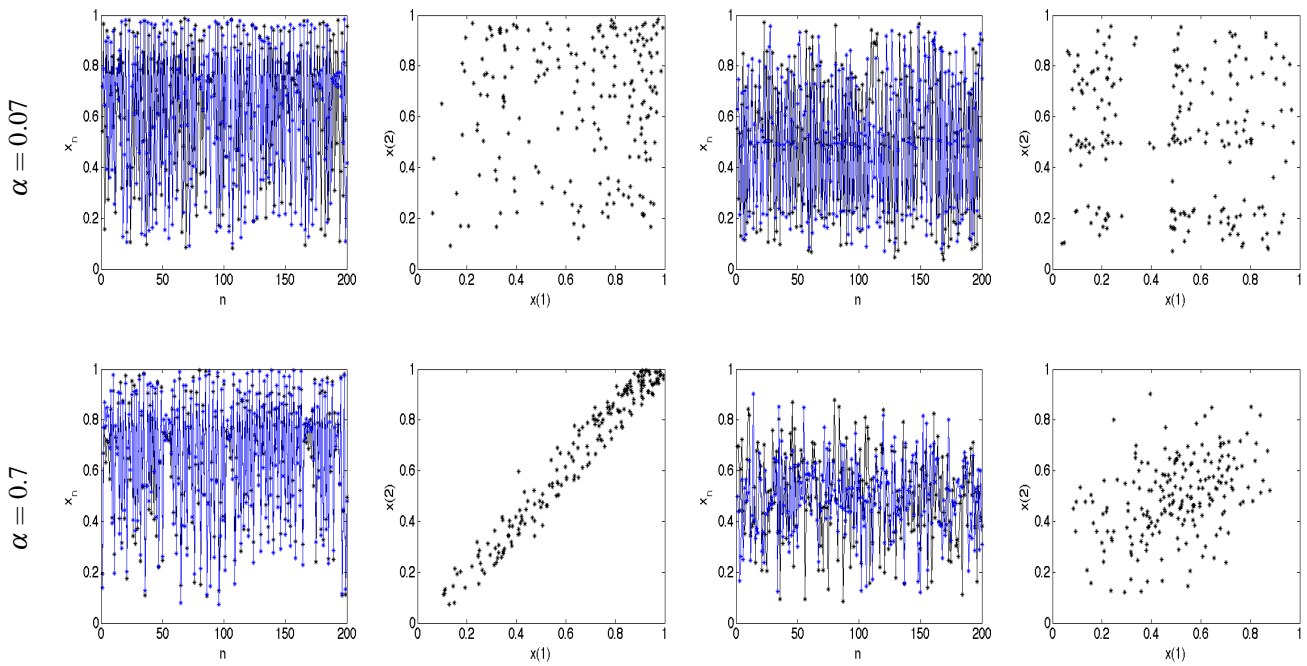


FIGURE 4.3 – time series windows for 2 logistic (first column) ; circle maps (third column) sharing a direct link under different coupling strengths α . The second and fourth columns show the corresponding phase spaces i.e. x_1 vs x_2 .

pour les connexions directes et indirectes sont mélangées pour former une courbe continue. Pour $\alpha = 0.7$, la courbe n'est pas continue mais il est pourtant impossible d'inférer correctement la topologie sous-jacente. Cela peut s'expliquer par le fait que les discontinuité ne divisent pas "clairement" les nœuds en deux ensembles bien distincts.

Dès lors, l'efficacité du MIR pour inférer la structure sous-jacente repose sur l'existence d'un saut dans ses valeurs ordonnées. Cela correspond à une différence entre les valeurs du MIR pour des connexions directes et les connexions indirectes. Lorsqu'une telle discontinuité existe, n'importe quelle valeur de seuil prise dedans permet d'inférer la topologie sous-jacente sans erreur. En fait, pour pouvoir avoir de "bonnes" discontinuités, il ne faut ni une trop petite force de couplage car il s'agit alors d'une situation "sans couplage", ni une trop grande force de couplage car ça crée une fausse topologie trop forte. En vérité, pour l'exemple choisi ici, il faut que α soit compris entre 0.03 et 0.15.

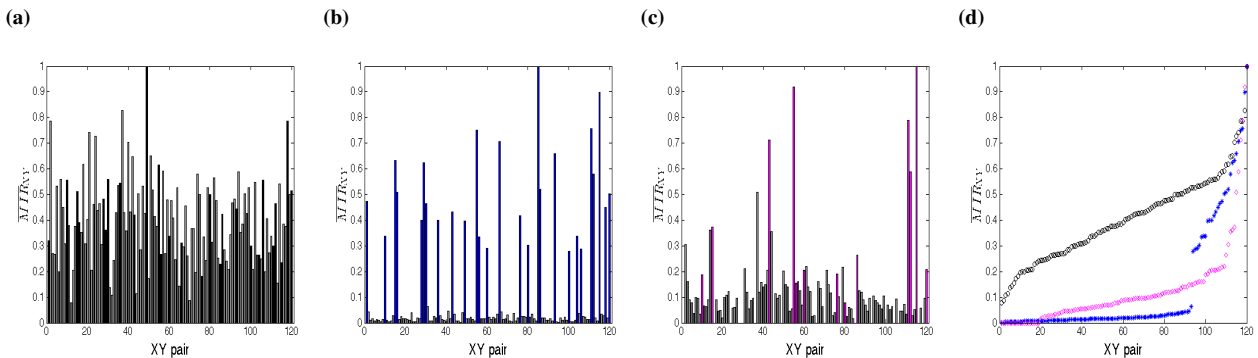


FIGURE 4.4 – (a)-(c) montrent les différentes valeurs du MIR pour $\alpha = 0, 0.07, 0.7$ respectivement. Les barres non grisées représentent les MIR entre les nœuds réellement connectés. Les différents MIR ordonnés (d) présentent des discontinuités qui correspondent (ou pas) aux seuils pour lesquels 100% d'inférence correcte est atteinte. En noir : $\alpha = 0$, en bleu : $\alpha = 0.07$ et en magenta : $\alpha = 0.7$.

4.2.3 Nombre de bins

Le calcul des probabilités repose sur le nombre de bins et il convient de bien choisir ce nombre de façon à obtenir des probabilités significatives. De façon évidente, un petit nombre de bins décroît la sensibilité tandis qu'un grand nombre provoque une perte de généralité. Bien que le calcul du $\overline{\text{MIR}}_{XY}$ soit basé sur différentes tailles de grilles, il serait bon d'essayer de déterminer un nombre minimal et maximal de bins (chaque bin a la même taille) par $\overline{\text{MIR}}_{XY}$ i.e. par grid ainsi qu'un nombre maximal de grid.

En ce qui concerne le nombre de bins, plusieurs propositions existent [62]. En particulier, la règle de Freedman-

Diaconis stipule que pour un échantillon X de n données, le nombre idéal de bins serait

$$n_{bins} = \frac{\max(x) - \min(x)}{2 \cdot Q \cdot n^{-1/3}}$$

avec Q l'aire de répartition interquartile de X [61]. Cette estimation ne requiert aucune hypothèse sur la distribution de l'échantillon. Lorsque nous supposons la normalité des données, nous pouvons utiliser la règle de Scott

$$n_{bins} = \frac{\max(x) - \min(x)}{3.49 \cdot \sigma_X \cdot n^{-1/3}}$$

avec σ_X une estimation de l'écart-type de X [50]. Pour ces deux propositions, le nombre idéal de bins est fonction de $n^{-1/3}$. Une dernière estimation vient de Sturges [52] qui s'exprime comme suit

$$n_{bins} = 1 + \log_2 n.$$

Malheureusement, cette dernière peut conduire à une sous-estimation du nombre de bins optimal mais semble très bien convenir lorsque la taille n est grande. Cependant, plusieurs auteurs semblent s'accorder sur le fait que pour des ensembles à valeurs réelles, des histogrammes de 5 à 20 bins suffisent amplement (alors que les estimations précédentes dépassent bien souvent 20).

En vue de tester ces différentes propositions, nous reprenons le cas de la figure 4.1a. Bien que le calcul de \overline{MIR}_{XY} se fasse sur base de plusieurs grids, nous commençons en envisageant un seul grid. Nous voyons donc bien que plus il y a de bins dans le grid, au plus les valeurs du \overline{MIR}_{XY} sont élevées (figure 4.5). Par contre, l'inférence n'est pas pour autant meilleure lorsque nous envisageons plus de bins.

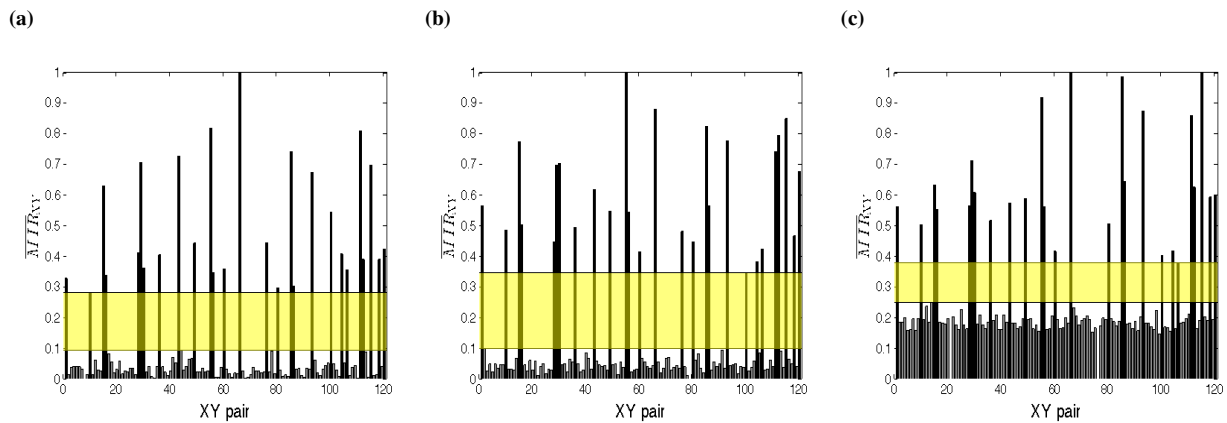


FIGURE 4.5 – Inférence de réseau pour la figure 4.1a avec un grid de (a) 5, (b) 10 et (c) 20 bins pour des séries temporelles de 20 000 données et $\alpha = 0.07$.

Pour rappel, le fait de considérer plusieurs grids se traduit dans le calcul du \overline{MIR}_{XY} par

$$\overline{MIR}_{XY} = \frac{\sum_i \hat{MIR}_{XY}(N_i)}{\max\{\sum_i \hat{MIR}_{XY}(N_i)\}}. \quad (4.23)$$

Le but recherché est donc de prendre des tailles de grids relativement différentes afin de "faire grandir les bons" en prenant de grandes tailles et de "diminuer les mauvais" en prenant des petites tailles. Dès lors, après différents essais, nous prenons trois tailles des grids : 5, 10 et $\text{ceil}(1 + \log_2(n))$ (figure 4.6). Dans nos exemples, le nombre de Freedman-Diaconis vaut environ 27, ce qui est de trop : cela est beaucoup plus gourmand en temps mais surtout dégrade la qualité de l'inférence. En effet, le temps de calcul augmente en fonction du nombre de bins, surtout lorsqu'il y en a plus que 15. Ce temps suit une loi exponentielle $f = 1.821 \cdot \exp(0.2645 \cdot n_{bins})$.

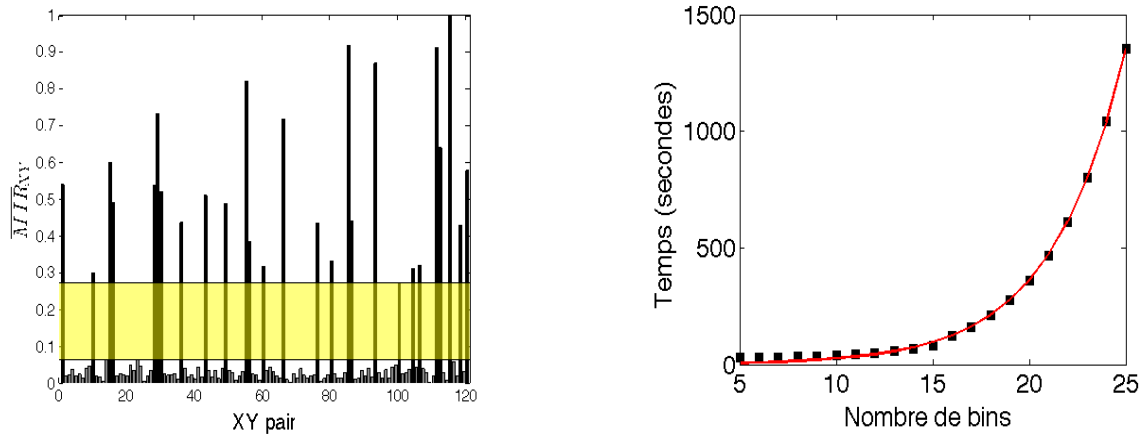


FIGURE 4.6 – (a) Inférence de réseau pour la figure 4.1a moyennée sur trois grids de 5, 10 et $\text{ceil}(1 + \log_2(n))$ bins pour des séries temporelles de 20 000 données et $\alpha = 0.07$; (b) Temps nécessaire pour calculer les différents \overline{MIR}_{XY} en fonction du nombre de bins. Les carrés noirs représentent les temps effectivement calculés tandis que la ligne rouge correspond à la fonction exponentielle $f = a * \exp(b * n_{bins})$ où $a = 1.821$ et $b = 0.2645$ (95% de confiance).

4.2.4 Longueur des séries temporelles et bruit

La figure 4.7 est générée afin d'avoir une vision plus générale de l'importance de la taille des séries temporelles et de la force de couplage α . Les seuils de reconstruction envisagés ici s'étendent de 0.1 à 0.7. Puisque le $\overline{MIR}_{XY} \in [0, 1]$, si le seuil égale 0, tous les liens possibles sont détectés : 120 pour la figure 4.1a et 15 pour 4.1b. Il n'y a donc que des "erreurs positives", des liens détectés qui ne le devraient pas, et le pourcentage d'erreur est $95/120 * 100 = 79.17$ pour la figure 4.1a et $8/15 * 100 = 53.33$ pour 4.1b. Au vu des graphes précédents (figures 4.5 et 4.6), les seuils supérieurs à 0.7 ne sont pas considérés car trop peu de liens seraient détectés.

Nous remarquons qu'avec un seuil supérieur à 0.3, nous ne parvenons plus à atteindre 100% d'inférence correcte. À partir de ce seuil, au plus nous montons, au plus le pourcentage d'inférence correcte semble se stabiliser autour de 70-80%. C'est parce que les "faux" liens ne sont plus détectés et seuls restent "les plus forts des bons" dont le \overline{MIR}_{XY} atteint presque 1. Notons également que, quel que soit le seuil envisagé, la dépendance vis-à-vis de la force de couplage est toujours bien présente tandis que celle vis-à-vis de la longueur des séries temporelles semble s'amenuiser à mesure que le seuil augmente. Cela signifie que la méthode fonctionne même pour des séries temporelles de petites tailles.

Au niveau des erreurs produites, nous avons, sans trop de surprise, que lorsque le seuil est bas, ce sont essentiellement des faux positifs tandis que les faux négatifs prennent le dessus lorsque le seuil augmente. En effet, un lien existant non détecté pour un seuil s_1 ne sera toujours pas détecté à un seuil $s_2 > s_1$. Lorsque la force de couplage est petite (inférieure à 0.05), les false positives ont tendance à être plus présentes. Peut-être car lorsque la force de couplage α est petite, le système est en fait composé de nœuds "non couplés" et par conséquent, la méthode "détecte tout". Une dernière observation : lorsqu'il y a quasi 100% d'inférence correcte, ce sont les false negatives qui sont présentes, i.e. nous ne détectons pas ce que nous devons mais ne détectons pas ce que nous ne devons pas : le "faux" liens sont bas ce qui est parfait mais certains "bons" liens aussi ce qui pose problème.

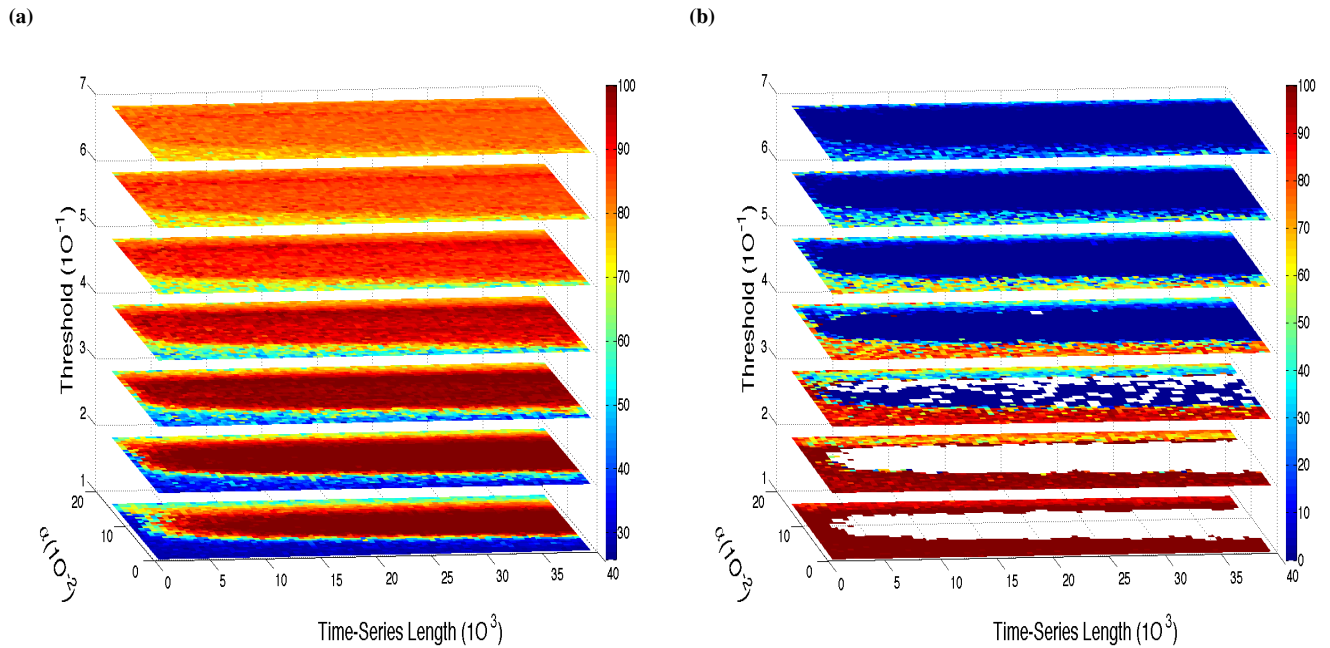


FIGURE 4.7 – Pourcentage (a) de reconstruction ; (b) d’erreurs positives - en fonction de la taille des séries temporelles, de la force de couplage et du seuil de reconstruction.

Ensuite, nous testons la méthode lorsque du bruit est ajouté. En particulier, nous considérons maintenant la dynamique suivante

$$f_{noise}(x_n, r) = f(x_n, r) + \gamma \cdot \sigma$$

où $\gamma \in [0, 1]$ représente la force du bruit et σ est un nombre aléatoire d’une loi normal de moyenne nulle et d’écart-type 1 i.e. $N(0,1)$. Dès lors, par les propriétés d’une loi normale, la force du bruit représente l’écart-type. La figure 4.8 montre le pourcentage de reconstructions correctes ainsi que le pourcentage d’erreurs positives en fonction du bruit et de la force de couplage α . Nous avons considéré des séries temporelles de 20 000 données. Nous parvenons à 100% d’inférence correcte pour quelques α mais uniquement lorsque le bruit n’est pas trop fort : $\gamma < 0.4$.

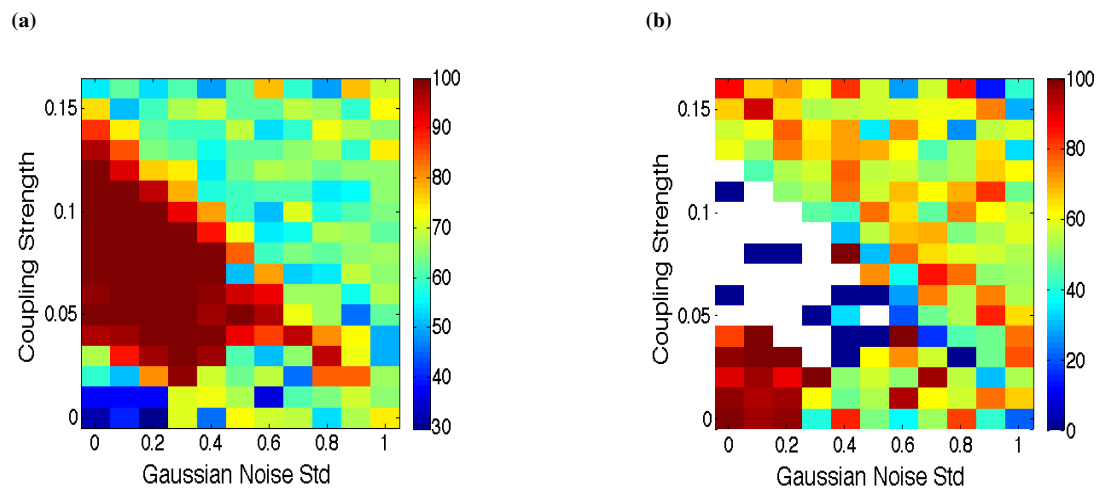


FIGURE 4.8 – Pourcentage (a) de reconstruction ; (b) d’erreurs positives - en fonction de la taille des séries temporelles, de la force de couplage pour un seuil de 0.25 lorsqu’un bruit est ajouté.

4.2.5 MI versus MIR

Nous confrontons ici le taux d’information mutuelle avec l’information mutuelle. Afin d’avoir une comparaison appropriée, nous normalisons de la même façon le MI. Nous appliquons nos méthodes sur le graphe (b) de la figure 4.1. Ce système contient deux clusters dont les nœuds possèdent une dynamique différente. Pour les trois premiers nœuds, la dynamique est la logistic map avec $r = 4$ tandis que pour les trois derniers, il s’agit de la logistic map composée trois fois avec elle-même i.e. $f(x^i) \equiv f \circ f \circ f(x^i)$ et $r = 3.9$. Nous avons pris des séries de 30 000 données et $\alpha = 0.05$.

Les différentes dynamiques génèrent des temps $T(N)$ différents pour que la corrélation disparaisse entre deux nœuds appartenant à des clusters différents.

La figure 4.9 reprend le pourcentage de reconstructions et d'erreurs positives en fonction de la longueur des séries temporelles, de la force de couplage α et du seuil de reconstruction. Nous remarquons que la méthode du MIR parvient mieux à distinguer les liens entre les nœuds que le MI lorsque nous sommes en présence de temps de "décorrélation" différents.. Cela est dû au fait que le MIR prend en compte le temps de décorrélation, contrairement au MI.

En revanche, lorsqu'il n'y a pas de différence dans les temps de "décorrélation", l'utilisation du MIR est moins justifiée. En effet, les différences observées entre le MI et MIR sont beaucoup plus ténues comme le montre la figure 4.10. Ces images tendent même à dire que le MI surpasse le MIR pour les α considérés. Nous remarquons également que le MI surpasse de beaucoup le MIR pour les courtes séries temporelles. Cela est dû au fait que le temps de décorrélation T est bien supérieur à la longueur N des séries ; le calcul des probabilités n'est donc pas significatif et la méthode du MIR produit de mauvais résultats.

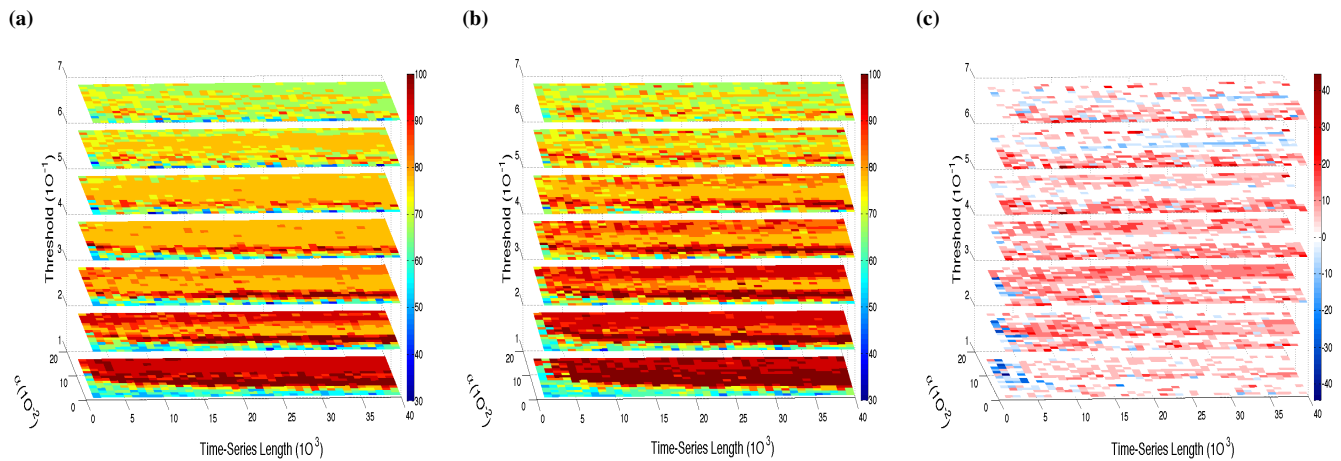


FIGURE 4.9 – Inférence de réseau pour la figure 4.1b sur base du (a) MI et (b) MIR. Les tableaux montrent le pourcentage de reconstruction pour différentes valeurs de force de couplage et de taille de données. Le panel (c) montre la différence entre MIR et MI : en rouge lorsque MIR surpasse MI, en bleu sinon.

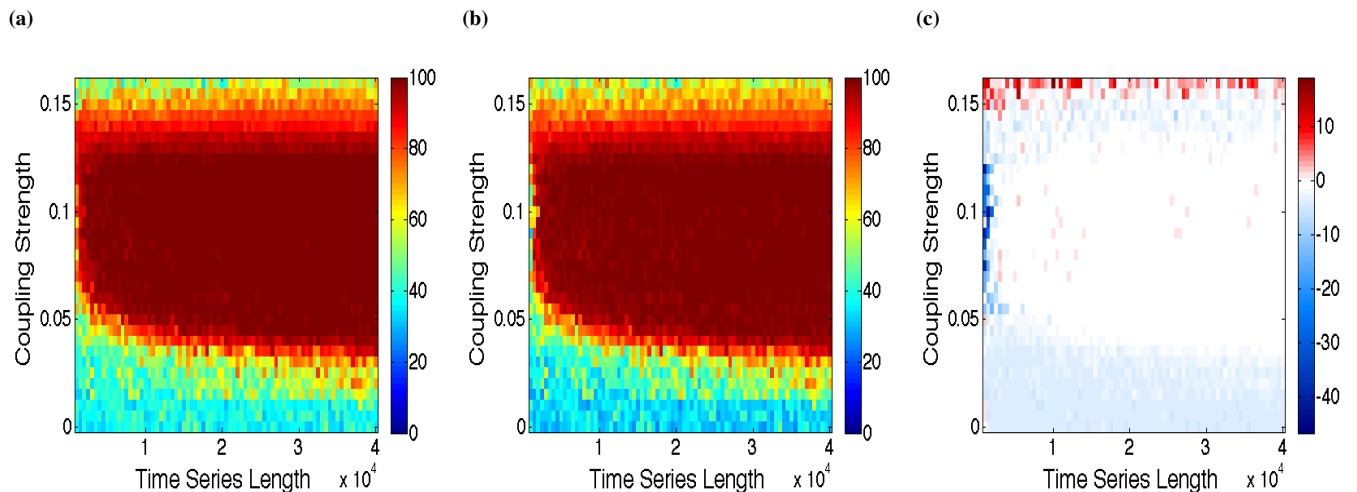


FIGURE 4.10 – Pourcentage d'inférence réussie par le (a) MI (b) MIR pour la figure 4.1a où tous les nœuds sont mis à jours grâce à la fonction logistique simple. Le seuil de reconstruction égale 0.25. Le panel (c) montre la différence entre MIR et MI : en rouge lorsque MIR surpasse MI, en bleu sinon.

4.2.6 Autre Histogramme

Jusqu'à présent, le calcul des probabilités se fait en partitionnant l'espace en N^2 cellules. Ces dernières sont le produit cartésien de N intervalles équidistants entre les valeurs minimales et maximales de deux signaux. Nous procédons maintenant à une autre partition de l'espace. Nous considérons toujours une partition formée de N^2 cellules qui sont maintenant le produit cartésien de N intervalles entre les valeurs minimales et maximales de deux signaux, choisis pour contenir environ n/N observations. En d'autres mots, nous ne divisons plus de façon équidistante (D) mais de façon

équiprobable (P). Avec une partition D , il se peut qu'il y ait beaucoup de cases vides, ce qui va fausser le calcul des probabilités. En envisageant une partition P , nous espérons réduire ces erreurs.

Pour comparer les deux méthodes, nous les appliquons sur la figure 4.1b avec des séries temporelles de 30 000 données. Les résultats sont obtenus en considérant plusieurs grids contenant N^2 cellules avec $N = 5, 10$ et $\text{ceil}(\log_2(n))$. Le panel 4.12c indique en rouge les cas où nous obtenons une meilleure inférence avec D qu'avec P . D peut être meilleur que P d'une plus grande quantité que P que D mais nous voyons surtout que P surpasse D dans plus de cas. De façon plus générale, la figure 4.11 reprend le pourcentage de cas où D et P se surpasse l'un l'autre en fonction du seuil de reconstruction envisagé. Ces chiffres sont calculé par *slice* comme à la figure 4.12a/b. Dans la majorité des cas, le pourcentage de bonne reconstruction est meilleur avec la méthode P . Nous regardons également les types d'erreurs produites à la figure 4.13. De façon surprenante, il existe quelques cas où les erreurs sont complètement opposées (dans la figure 4.12c caractérisé par une couleur foncée).

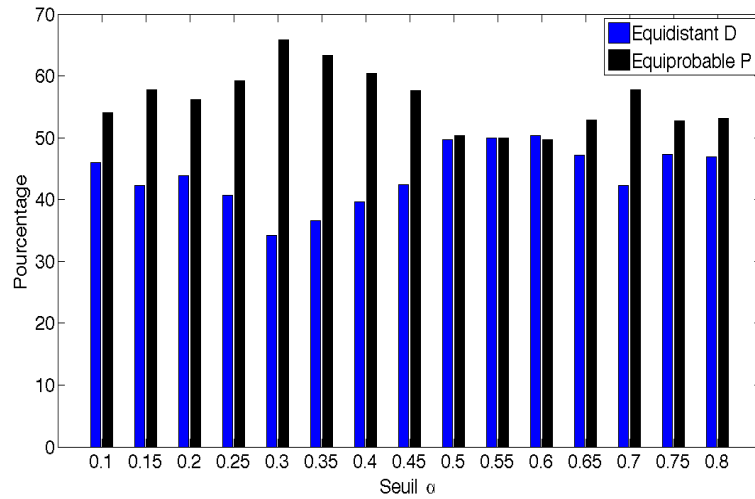


FIGURE 4.11 – Comparaison entre une partition équidistante et équiprobable. Les barres représentent le pourcentage de fois où chacune des deux méthodes surpasse l'autre pour un seuil α fixé.

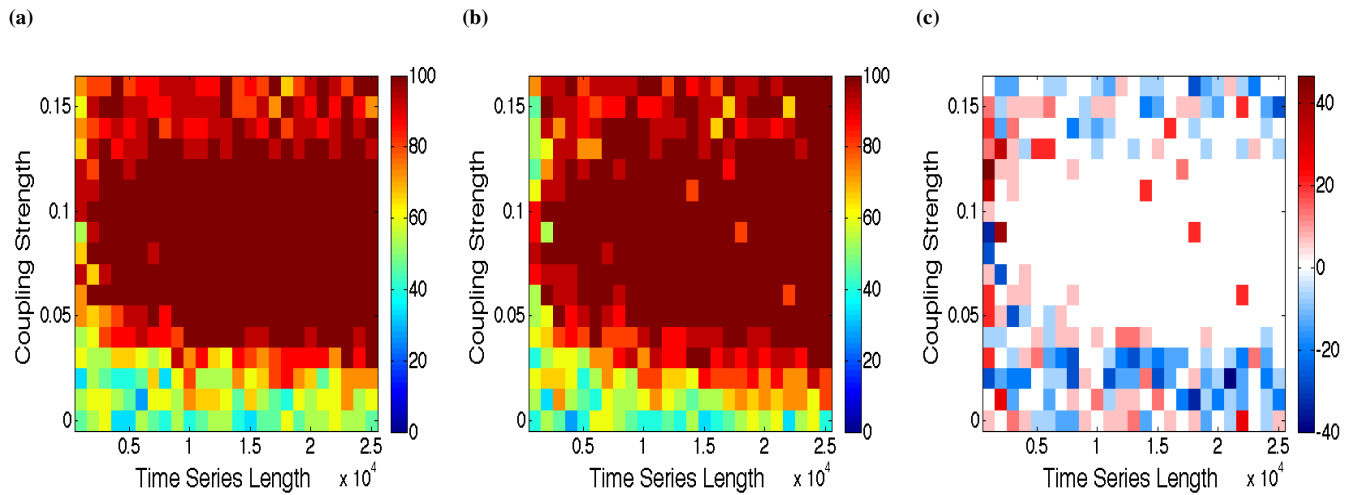


FIGURE 4.12 – Pourcentage d'inférence réussie par le MIR avec la méthode (a) D (b) P pour la figure 4.1b. Le seuil de reconstruction égale 0.2. Le panel (c) montre la différence entre équidistant et équiprobable : en rouge - quand équidistant surpasse équiprobable ; en bleu - sinon.

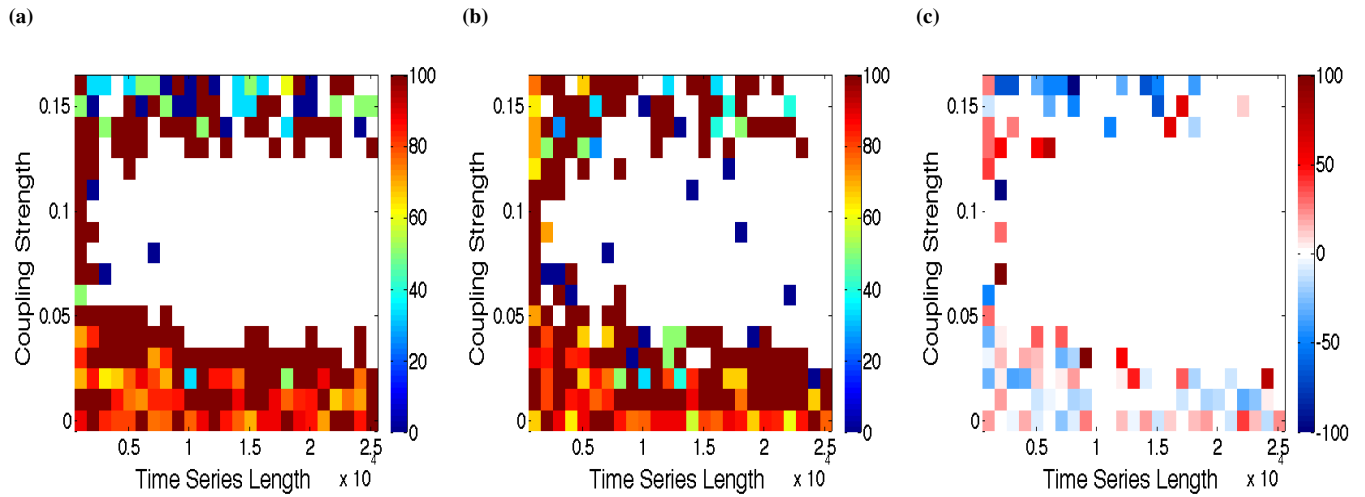


FIGURE 4.13 – Pourcentage d’erreur positive par le MIR avec la méthode (a) D (b) P pour la figure 4.1b. Le seuil de reconstruction égale 0.2. Sur le panel (c), en rouge lorsqu’il y a plus d’erreurs positives chez D que chez P .

4.3 Conclusion

Dans ce chapitre, nous avons introduit une mesure pour inférer la structure de réseau de systèmes complexes. Le taux d’information mutuelle MIR est une mesure d’information qui calcule l’information transférée par unité de temps entre différentes paires de composants d’un système complexe. Nous avons défini ici le $\overline{\text{MIR}}_{XY}$, une nouvelle normalisation de MIR, qui permet l’inférence de réseau. Nous avons montré que cette mesure était robuste pour l’inférence de réseau lorsque les séries temporelles sont de petite taille, en présence de bruit ainsi que pour les systèmes avec différentes forces de couplage. En outre, MIR et $\overline{\text{MIR}}_{XY}$ font intervenir le temps pour que la corrélation disparaisse et donc, sont deux mesures appropriées pour inférer correctement la topologie d’un réseau avec différentes échelles de temps. Dans ce cas, nous avons également montré que $\overline{\text{MIR}}_{XY}$ surpassait l’information mutuelle MI bien que ces deux mesures soient suffisamment robustes lorsqu’une seule échelle de temps était présente dans le réseau.

Chapitre 5

Dynamical Cluster Index

Nous introduisons dans ce chapitre la méthode du Dynamical Cluster Index qui permet d'identifier des sous-ensembles pertinents de variables utiles à la compréhension de l'organisation d'un système. Nous entendons par sous-ensemble pertinent un sous-ensemble dont les variables ont une forte interaction et intégration entre elles et qui sont séparées du reste du système. Ce chapitre se base sur [55]. Les auteurs de cet article insistent tout de même sur le fait que la liste de sous-ensembles obtenus est à prendre avec précaution, en fonction du cas auquel nous l'appliquons. En effet, nous ne pouvons affirmer avec certitude que ces sous-ensembles ainsi trouvés correspondent effectivement à des sous-ensembles "importants" dans l'organisation du système.

5.1 Aspects théoriques

Le Cluster Index est un indice basé sur l'information et est défini comme étant le rapport entre deux autres mesures, l'intégration et l'information mutuelle. Soient un système U composé de K éléments et un sous-ensemble S de $k < K$ éléments. Le cluster index $C(S)$ se définit comme

$$C(S) = \frac{I(S)}{M(S; U - S)} \quad (5.1)$$

où $I(S)$ mesure l'indépendance statistique des k éléments de S

$$I(S) = \sum_{s \in S} H(s) - H(S) \quad (5.2)$$

et $M(S; U - S)$ représente l'information mutuelle entre S et le reste du système

$$M(S; U - S) \equiv H(S) + H(S|U - S) = H(S) + H(U - S) - H(S, U - S) \quad (5.3)$$

avec $H(X|Y)$ l'entropie conditionnelle et $H(X, Y)$ l'entropie jointe. Lorsque $M(S; U - S)$ s'annule, la valeur du cluster index n'est pas définie ; cela signifie que l'ensemble S est totalement séparé du reste du système et requiert une étude séparée.

Notons également que cet indice dépend de la taille de l'ensemble S ce qui n'est pas pratique pour comparer des ensembles de tailles différentes. Pour pallier à ce problème, nous normalisons grâce à un système homogène U_h . Ce système homogène est généré de manière aléatoire selon la probabilité de chaque état mesuré dans le système original U . En d'autres mots, nous calculons la fréquence de chaque symbole dans les différentes séries et générons de nouvelles séries aléatoires pour lesquelles chaque symbole apparaît avec une même probabilité que dans la série originale associée. Ce système U_h ainsi formé possède donc des propriétés statistiques du système U mais ne contient (normalement) pas de clusters. Il existe différentes façons de générer ces systèmes, comme expliqué dans la suite. Pour chaque taille de sous-ensemble du système U_h , nous calculons l'intégration moyenne $\langle I_h \rangle$ ainsi que l'information mutuelle moyenne $\langle M_h \rangle$. Ces dernières permettent d'obtenir un cluster index normalisé pour n'importe quel sous-ensemble S

$$C'(S) = \frac{I(S)}{\langle I_h \rangle} \bigg/ \frac{M(U, S - U)}{\langle M_h \rangle}. \quad (5.4)$$

Enfin, nous pouvons également calculer un indice statistique T_c permettant d'évaluer l'importance statistique du cluster index

$$T_c(S) = \frac{C'(S) - \langle C'_h \rangle}{\sigma(C'_h)} = \frac{vC(S) - v\langle C_h \rangle}{v\sigma(C_h)} = \frac{C(S) - \langle C_h \rangle}{\sigma(C_h)} \quad (5.5)$$

où $\langle C'_h \rangle$, $\sigma(C'_h)$, $\langle C_h \rangle$ et $\sigma(C_h)$ représentent respectivement la moyenne et l'écart-type des ensembles, de même taille que S , de cluster indices normalisés et non normalisés du système homogène et $v = \langle M_h \rangle / \langle I_h \rangle$.

En vue de déterminer les sous-ensembles pertinents, il faudrait en réalité considérer tous les sous-ensembles possibles. En pratique, cela n'est faisable que pour des systèmes de petites tailles. Pour pouvoir considérer des plus grands ensemble, nous procédons ici à une recherche aléatoire : pour chaque taille de sous-systèmes, nous ne tirons qu'une fraction f de toutes les possibilités : $f \times \frac{N!}{k!(N-k)!}$. Évidemment, un tel échantillonnage aléatoire est susceptible de manquer des sous-ensembles dont leur DCI est grand. En vue de minimiser ce problème, nous ajoutons une simple recherche heuristique. Plus précisément, les échantillons de tailles k sont composés non seulement d'échantillons aléatoires mais également des échantillons formés à partir du meilleur sous-ensemble (dont le DCI est le plus haut) de taille $k - 1$ auquel nous ajoutons les voisins. Ceci est motivé par le fait que les ensembles dont le DCI est haut sont composés de sous-ensembles ayant eux-mêmes un haut DCI. Nous générons ainsi des échantillons de taille 2 jusque $N - 1$ et nous les classons en fonction de leur T_c .

5.2 Quelques résultats

Dans cette section, nous testons cette méthode sur différents systèmes artificiels. Dans un premier temps, nous considérons des réseaux booléens. Ensuite nous envisageons une dynamique *leader-follower*. Finalement, nous essayons la méthode avec le réseau et la dynamique de la figure 4.1b décrits précédemment.

5.2.1 Réseaux booléens

Les réseaux booléens (BNs) utilisés sont illustrés à la figure 5.1. Le premier BN est constitué des deux ensembles indépendants tandis que dans le second, les composants interagissent entre eux. Quant au dernier, il est l'union des deux précédents.

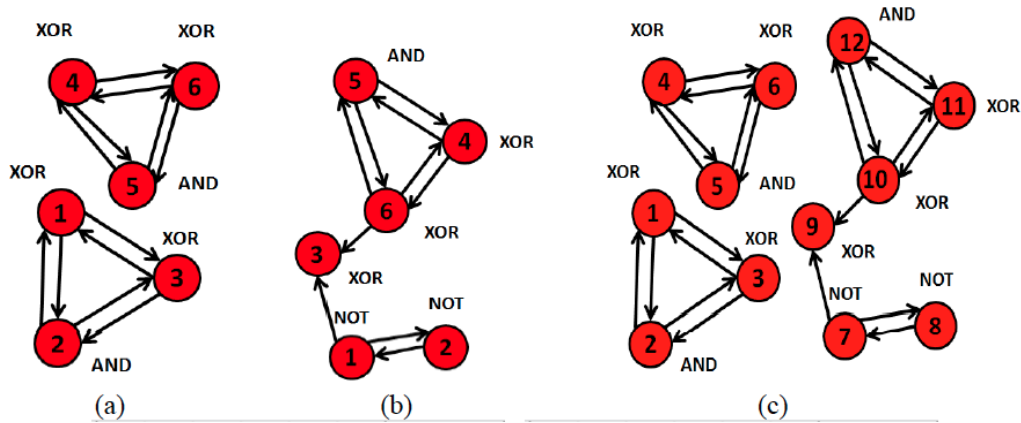


FIGURE 5.1 – (a) Réseaux booléens indépendants ; (b) Réseau interdépendant ; (c) Système composé de l'union des deux réseaux précédents. La fonction booléenne associée à chaque nœud est indiquée à son côté [55].

Nous comparons les rankings obtenus à partir de 20 systèmes homogènes (voir plus bas). Pour chaque système de la figure 5.1, nous avons posé $f = 0.5$ et considéré des séries temporelles de 500 données (table 5.1).

Système 1			Système 2			Système 3		
Ranking	p	RS	Ranking	p	RS	Ranking	p	RS
1	0.9	4,5,6	1	1	2,3,4,5,6	1	0.8	8,9,10,11,12
2	0.7	1,2,3	2	0.5	4,5,6	2	0.7	4,5,6
3	0.6	3,4,5,6	3	0.33	2,3,5	3	0.5	7,8,9,10,11,12
4	0.6	2,4,5,6	4	0.3	2,3,4,5	3	0.5	1,2,3
5	0.55	1,4,5,6	5	0.53	3,4,5,6	4	0.5	1,2,3
						5	0.43	8,9,11

TABLE 5.1 – Résultats des 3 systèmes. Pour chaque ranking, le sous-ensemble (RS, *relevant subset*) obtenu le plus fréquemment est indiqué, accompagné de sa fréquence d'apparition.

Nous pouvons également conforter ces résultats en s'intéressant à la probabilité qu'ont les sous-ensembles d'occuper une des cinq premières places (table 5.2).

En vue de nettoyer un peu les données récoltées, nous pouvons nous limiter aux sous-ensembles disjoints ou overlapping. En d'autres mots, si un sous-ensemble est entièrement contenu dans un autre, nous retenons uniquement celui ayant le plus haut DCI (table 5.3).

Nous avons également testé la méthode en faisant varier la longueur des séries temporelles : 1, 2, 3, 10, 20, 30 et 40 fois la taille de la série de base. Mais étant donné que nous atteignons très rapidement un attracteur, il n'y a pas de différence dans les résultats des rankings. En revanche, nous remarquons que les valeurs des Tcs diffèrent en fonction de la longueur des séries temporelles ; ce sont principalement les Tcs correspondant aux meilleurs RS qui fluctuent (figure 5.4).

Système 1		Système 2		Système 3	
p	RS	p	RS	p	RS
0.95	1,2,3,6	1	2,3,4,5,6	1	7,8,9,10,11,12
0.9	1,2,3	0.9	2,3,4,5	0.95	1,2,3
0.9	1,4,5,6	0.87	2,3,5,6	0.9	4,5,6
0.85	4,5,6	0.73	4,5,6	0.9	1,2,3,5
0.7	3,4,5,6	0.7	2,3,5	0.65	7,8
0.4	1,3	0.4	1,2,3,5	0.55	8,9,10,11,12
0.4	1,4,5,6	0.33	1,2	0.4	1,3
0.1	1,2,3,4	0.033	4,6	0.25	7,8,9,10,11
0.05	4,6	0.033	2,3,4,6	0.2	8,9,11
0.05	1,2,5,6			0.1	1,2,3,12

TABLE 5.2 – Résultats des 3 systèmes. Pour chaque sous-ensemble (RS, *relevant subset*), nous notons la probabilité qu'a ce dernier d'apparaître dans les cinq premiers rangs.

Système 1			Système 2			Système 3		
Ranking	p	RS	Ranking	p	RS	Ranking	p	RS
1	0.9	4,5,6	1	1	2,3,4,5,6	1	1	8,9,10,11,12
2	0.7	1,2,3	2	0.6	1,2,3,4	2	0.9	7,8,9,10,11
3	0.6	1,3,5	3	0.5	1,2,3,5	3	0.8	1,2,3
4	0.3	2,6	4	0.5	1,2,5	4	0.7	4,5,6
5	0.4	1,3,6	5	0.5	1,4,5,6	5	0.6	4,10,11,12

TABLE 5.3 – Résultats des 3 systèmes après nettoyage. Pour chaque sous-ensemble (RS, *relevant subset*), nous notons la probabilité qu’a ce dernier d’apparaître dans les cinq premiers rangs.

Comme mentionné précédemment, les résultats sont obtenus à partir de 20 systèmes homogènes. Cette valeur a été choisie car il s’agit d’un bon compromis entre temps et différence entre significatif et insignifiant. En réalité, le nombre de systèmes homogènes n’intervient pas dans la bonne détection, i.e. les sous-ensembles devant réellement être détectés (testé sur le cas 1 et 2) mais impacte les valeurs des Tcs (figures 5.2 et 5.3). Cependant, il ne semble pas y avoir de relation claire entre ces valeurs et le nombre de systèmes homogènes. Nous avons également généré nos systèmes homogènes de différentes façons. Les résultats repris ici sont en fait obtenus en permutant les données sur l’ensemble du système U : l’entropie de chaque élément est par conséquent susceptible de changer. Nous avons également testé en permutant uniquement au sein d’une même variable de telle façon que l’entropie de chaque élément ne change pas mais les résultats obtenus étaient identiques. Par conséquent et afin de ne pas alourdir inutilement le rapport, nous ne reprenons pas ces résultats.

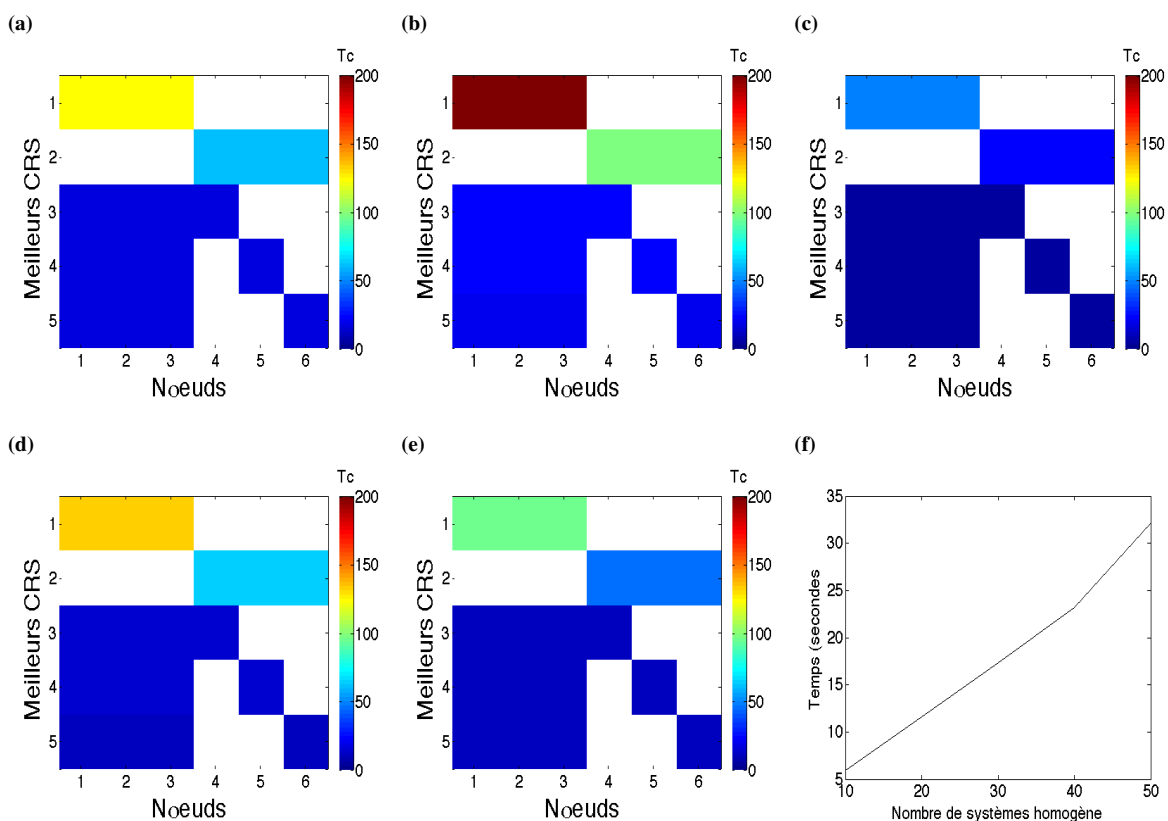


FIGURE 5.2 – Tcs en fonction du nombre de systèmes homogènes générés (a) 10, (b) 20, (c) 30, (d) 40 et (e) 50 pour le système 1. Les séries temporelles sont composées de 1000 données. Les nœuds formant un CRS sont colorés. Nous retrouvons toujours les mêmes meilleurs CRS mais leur Tc diffèrent en fonction du nombre de U_h . (f) représente le temps requis pour le calcul des Tcs.

5.2.2 Dynamique *leader-follower*

Nous appliquons ici la méthode sur un simple modèle comparable à un modèle *leader-follower*. Plus précisément, ce système est composé de vecteurs binaires indépendants de n variables $x = [x_1, x_2, \dots, x_n]$ et est généré de la manière

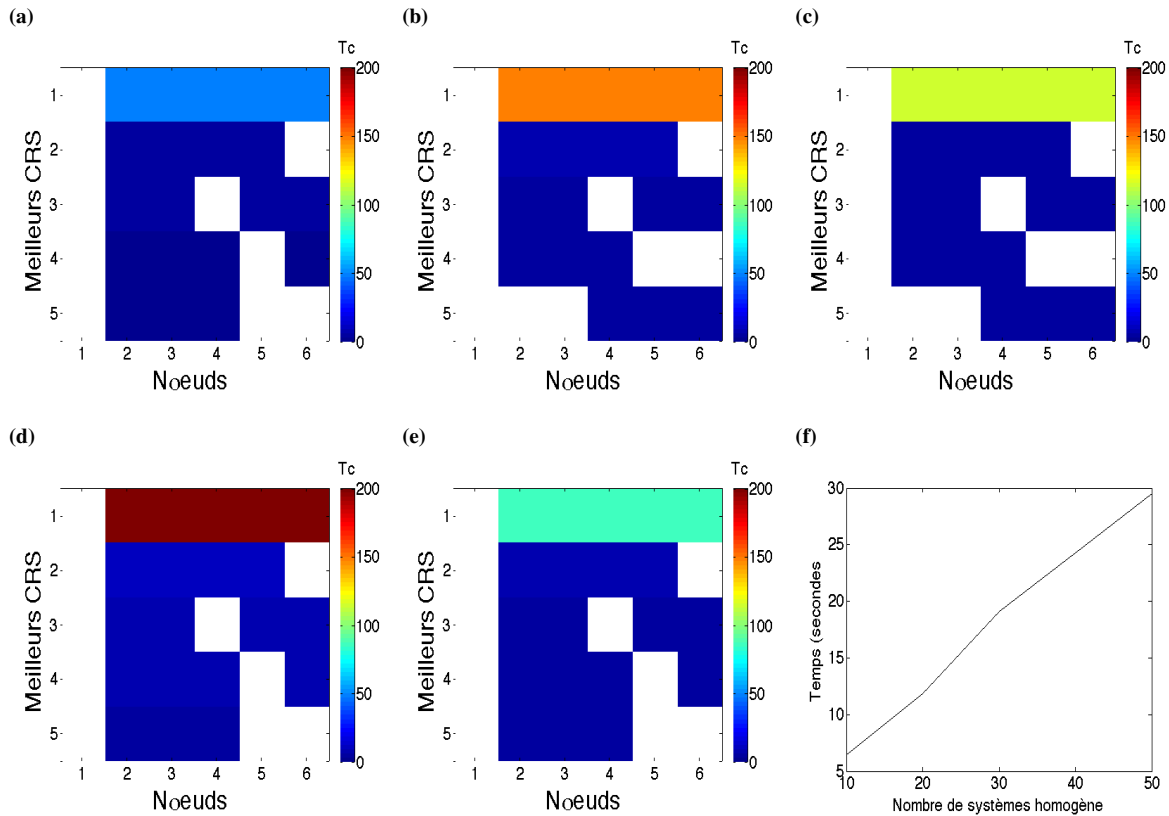


FIGURE 5.3 – T_c s en fonction du nombre de systèmes homogènes générés (a) 10, (b) 20, (c) 30, (d) 40 et (e) 50 pour le système 2. Les séries temporelles sont composées de 1000 données. Les nœuds formant un CRS sont colorés. (f) représente le temps requis pour le calcul des T_c s.

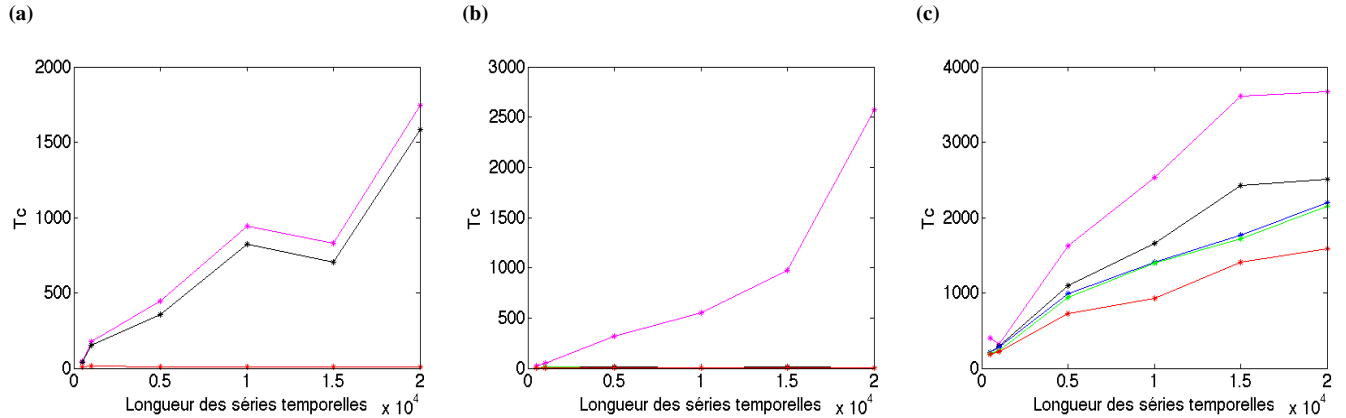


FIGURE 5.4 – Evolution des T_c s en fonction de la longueur des séries temporelles du (a) Cas 1 ; (b) Cas 2 ; (c) Cas 3. Les T_c s sont calculés sur base de 20 systèmes homogènes. Les cinq meilleurs T_c s sont représentés par ordre décroissant : magenta, noir, bleu, vert, rouge.

suivante :

- les variables sont divisées en deux sous-ensembles $G1 = [x_1, \dots, x_k]$ et $G2 = [x_{k+1}, \dots, x_n]$;
- x_1 est appelé le *leader* et nous lui assignons arbitrairement la valeur 0 ou 1 ;
- x_2, \dots, x_k sont les *followers* et sont tels que $x_i = x_1$ avec probabilité $1 - p_{noise}$, $x_i \in \{0, 1\}$ aléatoirement avec probabilité p_{noise} , $i = 2, \dots, k$;
- les valeurs des éléments de $G2$ sont une copie d'un élément aléatoire de $G1$ avec probabilité p_{copy} ou bien une valeur aléatoire dans $\{0, 1\}$ avec probabilité $1 - p_{copy}$.

L'avantage de ce système est que nous pouvons jouer avec les paramètres p_{noise} et p_{copy} en vue de changer l'intégration parmi les éléments de $G1$ et l'information mutuelle entre $G1$ et $G2$. Les différents cas possibles sont illustrés à la figure 5.5. Ces figures sont obtenues en considérant des systèmes de 1000 vecteurs de tailles $n = 10$ avec $k = 3$ (Nous obtenons les mêmes allures de graphes en considérant d'autres valeurs pour n et k). L'intégration entre les éléments de $G1$ est

une fonction décroissante de p_{noise} et, sans surprise, indépendante du p_{copy} . Quant à l'information mutuelle entre $G1$ et $G2$, nous remarquons qu'elle croît en fonction des deux probabilités p_{noise} et p_{copy} mais plus rapidement en fonction de p_{copy} , qui accroît la corrélation entre $G1$ et $G2$. La raison pour laquelle l'information mutuelle croît également, même si plus faiblement, avec p_{noise} est que cette dernière accroît l'aléatoire parmi les éléments de $G1$, qui se comportent de la même façon dans $G2$.

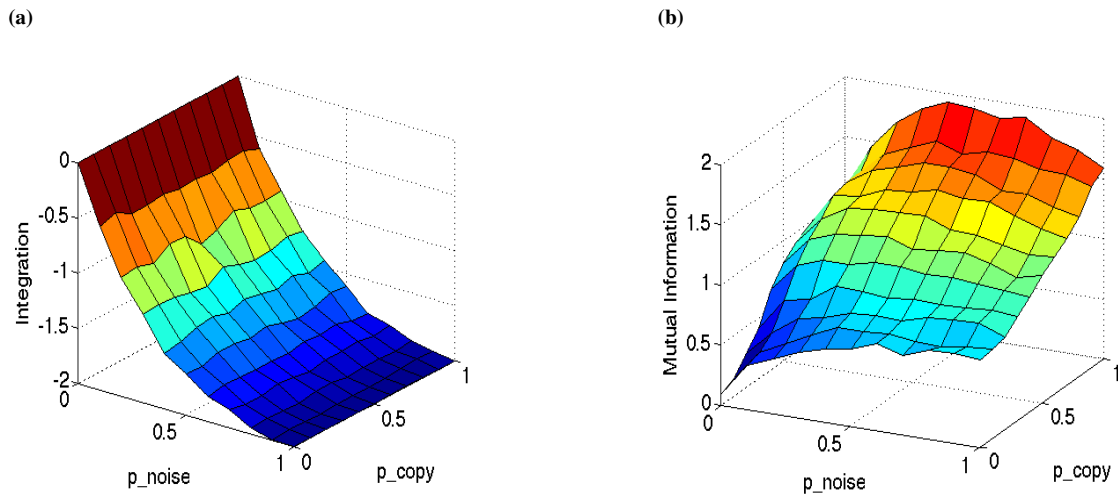


FIGURE 5.5 – (a) Intégration parmi les variables du sous-ensemble $G1$; (b) Information mutuelle entre les deux sous-ensembles $G1$ et $G2$ - en fonction des valeurs de p_{noise} et p_{copy} .

Notons tout de même le cas particulier qui survient lorsque l'information mutuelle est quasi nulle. Cela signifie que $G1$ est presque complètement indépendant de $G2$. Comme mentionné précédemment, nous ne nous préoccupons pas de ce cas. En revanche, nous nous intéressons aux cas pour lesquels les deux quantités I et MI sont significativement différentes de zéro. Nous regardons pour chacune des valeurs de p_{noise} et $p_{copy} \in \{0, 0.1, \dots, 1\}$ les cas où la méthode est capable d'identifier les deux sous-ensembles $G1$ et $G2$. Cela est illustré à la figure 5.6 où les zones rouges foncées dénotent les cas où le groupe $G1$ est identifié avec le plus grand Tc . La méthode est capable de détecter avec succès l'ensemble $G1$ jusqu'à $p_{noise} = 0.9$. En outre, même lorsque p_{copy} est élevé (pour des petites valeurs de p_{noise}), l'ensemble $G1$ est bien détecté. Ces résultats sont obtenus en générant 10 systèmes homogènes U_h .

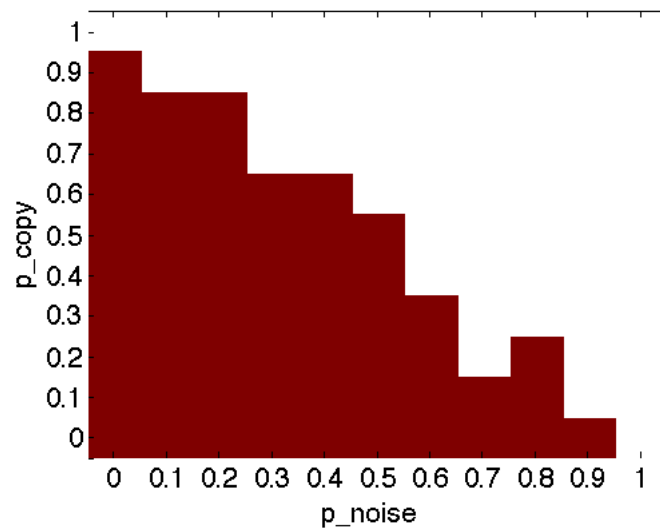


FIGURE 5.6 – Matrice représentant les cas où l'ensemble $G1$ est identifié par la méthode DCI avec le plus grand indice Tc . Les cases rouges foncées dénotent les cas bien détectés.

5.2.3 Modèle logistique

Nous considérons le système donné par la figure 4.1b. Nous nous attendons à ce que la méthode détecte les deux sous-ensembles composés des nœuds 1-2-3 et 4-5-6 mais également tous les sous-ensembles de taille 2 appartenant à ces deux sous-ensembles ; ceci est motivé par le fait que des paires peuvent plus facilement être intégrées que les triplets. Cela est confirmé par la figure 5.7. Pour un découpage à 5 bins, il y a des sous-ensembles "parasites" dans le top8 au ranking

5, 6 et 7. Cela signifie qu'il n'y a pas assez de bins. Pour un découpage de 8 à 15 bins, nous remarquons un classement similaire : d'abord les nœuds composant le sous-ensemble 1-2-3 et ensuite ceux formant 4-5-6. Notons que l'estimation de Sturges donne dans ce cas $nbins = 1 + \log_2 20000 = 14$. Les résultats sont obtenus à partir de 20 systèmes homogènes et d'une force de couplage $\alpha = 0.07$.

Observons également l'évolution des Tcs values en fonction du nombre de bins et de la longueur des séries temporelles (figure 5.8). Lorsque nous faisons varier le nombre de bins, les valeurs des Tcs décroissent mais tendent à se stabiliser à partir de 11 bins. La décroissance est surtout due à une augmentation des valeurs de l'information mutuelle. Notons également que lorsque le nombre de bins augmente, $\langle C_h \rangle$ augmente alors que $\sigma(C_h)$ décroît et se stabilise. La figure 5.8 suggère également que les Tcs croissent en fonction de la longueur des séries temporelles. Cela est principalement causé par une diminution des valeurs de l'information mutuelle, celles de l'intégration restent relativement constantes.

Afin d'avoir une vision plus large, nous faisons varier la force de couplage α ainsi que la longueur des séries temporelles (figure 5.10) et regardons le pourcentage de détection des 8 meilleurs sous-ensembles. Nous ajoutons également un bruit afin de tester la robustesse de la méthode mais remarquons que la méthode ne détecte les bons ensembles que lorsque le bruit est faible.

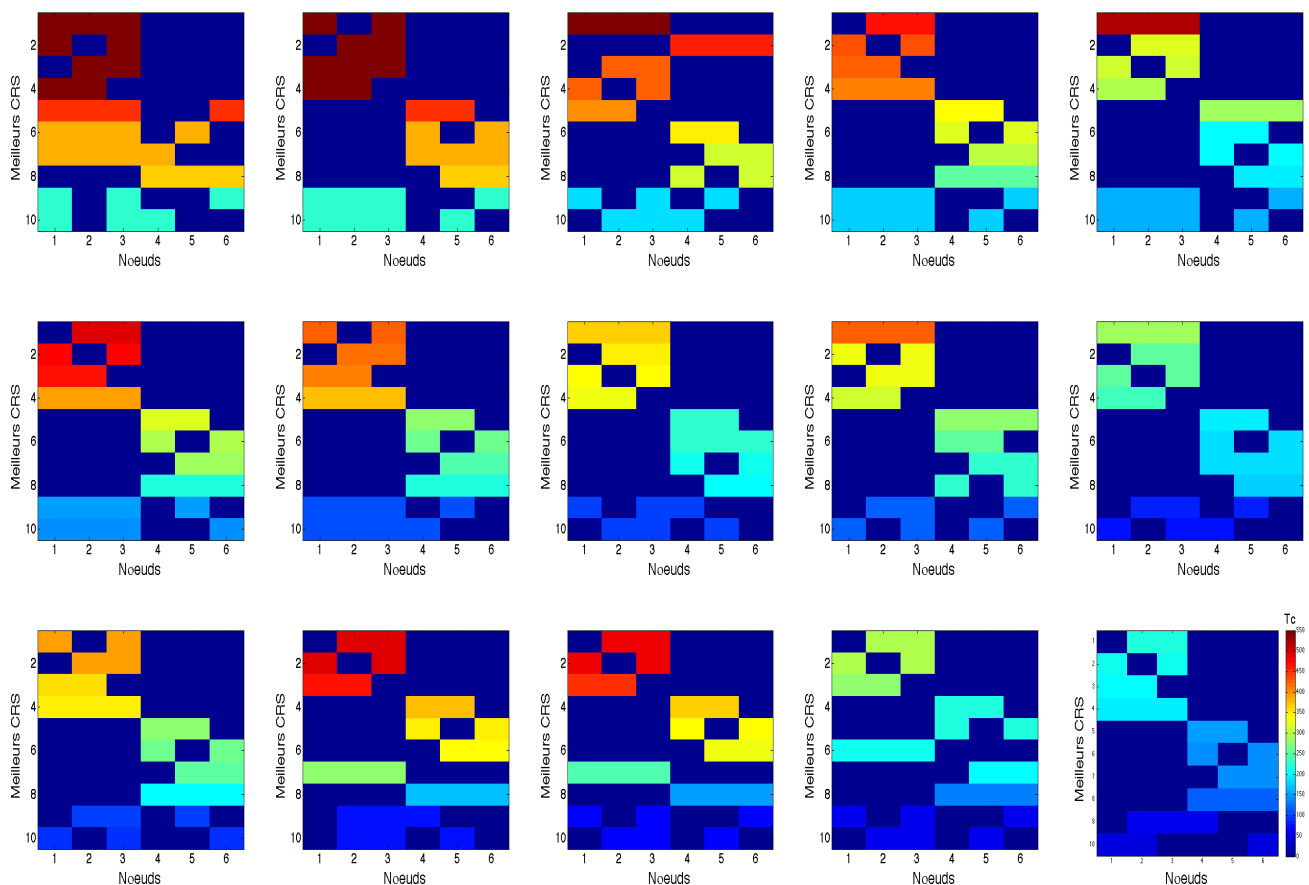


FIGURE 5.7 – Les 10 meilleurs CRS (en fonction de leur T_c) pour 5 à 19 bins (par pas de 1) pour des séries temporelles de 20 000 données. Les séries suivent le modèle logistique tel que décrit à la figure 4.1b.

5.3 Conclusion

Dans ce chapitre, nous avons introduit une méthode permettant d'identifier des sous-ensembles pertinents de variables utiles à la compréhension d'un système dynamique. Cette méthode ne requiert aucune connaissance de relations entre les variables du système ; elle repose uniquement sur des observations de valeurs dans le temps. Elle a également l'avantage de prendre compte des corrélations multiples et non uniquement des corrélations de paires. Nous avons pointé le fait que les systèmes homogènes U_h pouvaient être générés de différentes façons mais que pour nos exemples considérés, il n'y avait pas de différences. Nous avons aussi évoqué le fait que pour trouver les ensembles de variables recherchés, il faut parcourir tous les sous-ensembles possibles, ce qui n'est pas faisable en pratique. Nous avons utilisé une recherche aléatoire ainsi qu'une (trop) simple recherche heuristique. Il serait bon d'améliorer ce dernier point avec, par exemple, un algorithme génétique comme suggéré dans [55]. Nous n'avons considéré que trois exemples ; ils ont cependant permis de montrer les limites de la méthode lorsque les signaux étaient pollués par du bruit.

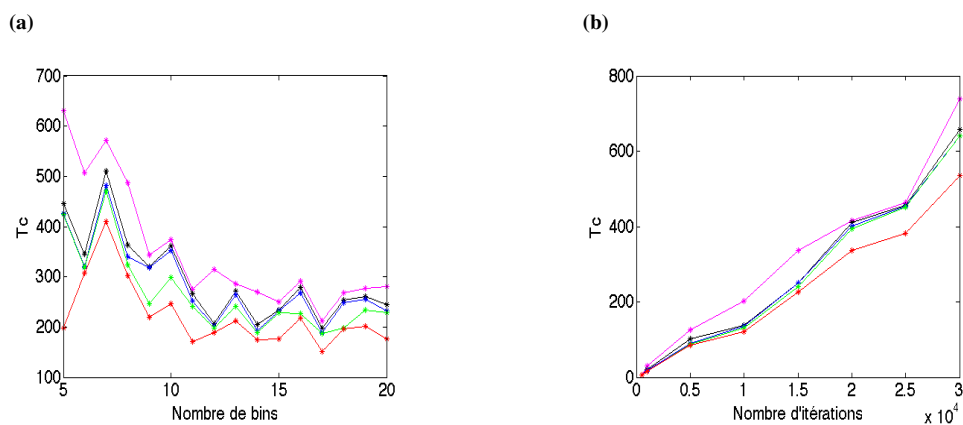


FIGURE 5.8 – Évolution des indices T_{cs} en fonction (a) du nombre de bins (20 000 données) ; (b) de la longueur des séries temporelles (14 bins). Les séries suivent le modèle logistique tel que décrit à la figure 4.1b.

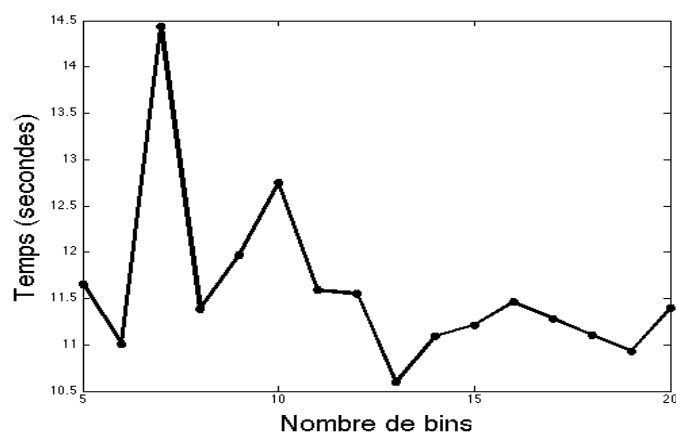


FIGURE 5.9 – Temps requis pour calculer les indices T_{cs} avec des séries composées de 20 000 données. Les séries suivent le modèle logistique tel que décrit à la figure 4.1b.

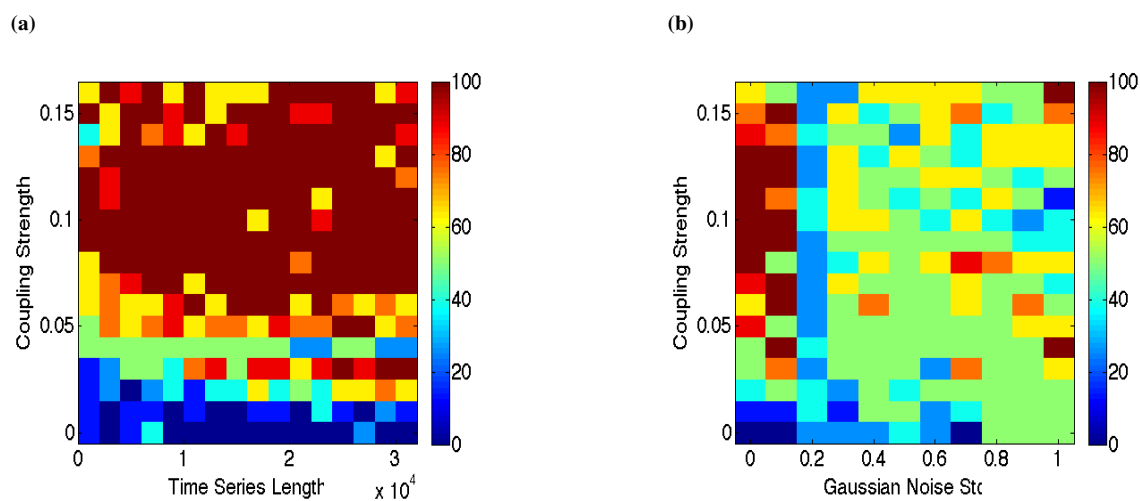


FIGURE 5.10 – Pourcentage de détection réussie des 8 meilleurs CRS en considérant 14 bins (a) sans ; (b) avec un bruit ajouté. Pour le panel (b), les séries temporelles sont composées de 30 000 données.

Chapitre 6

Application

Dans ce dernier chapitre, nous tentons d'appliquer les méthodes vues dans les chapitres précédents sur deux ensembles de données. Le premier reprend le nombre de personnes infectées (déclarées) par la dengue dans la municipalité de Rio de Janeiro pour l'année 2007. Le second comprend 90 mesures relatives au cerveau sur 140 unités de temps.

6.1 Virus de la dengue

Dans cette partie, nous appliquons quelques concepts vu précédemment sur des données relatives au virus de la dengue. Ces données ont été fournies par Stefanella Boatto et reprennent le nombre d'individus infectés par le virus de la dengue dans la municipalité de Rio de Janeiro. L'aménagement du territoire de Rio de Janeiro est divisé en 5 zones de planification, elles-mêmes divisées en districts (figure 6.1). Les données sont reprises pour chaque district semaine par semaine et couvrent l'année 2007. Dans la suite, nous renommons les *área de planejamento* 1, 2.1, ... par *district* 1, 2, ... 10.

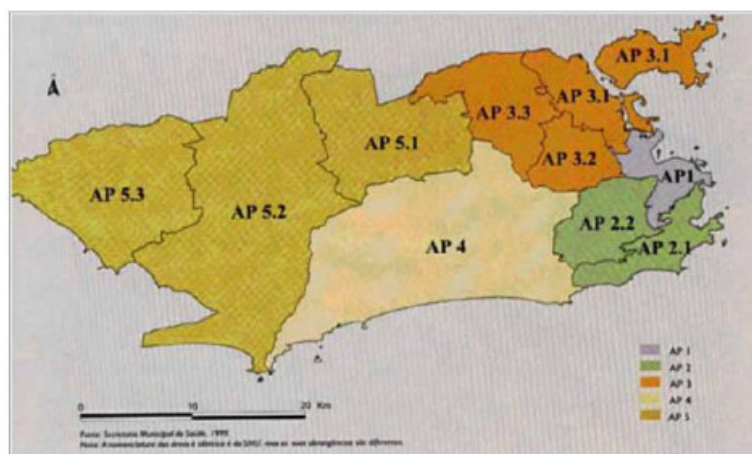


FIGURE 6.1 – Arrangement du territoire de la municipalité de Rio de Janeiro [16].

Rio de Janeiro

Nous traitons les données pour l'entièreté de la municipalité de Rio de Janeiro. Afin de voir l'évolution temporelle, nous appliquons l'algorithme de visibilité horizontale sur chaque fenêtre temporelle. En particulier, nous avons une série temporelle de $N=52$ données $\{x_1, x_2, \dots, x_{52}\}$. Nous choisissons une taille de fenêtre s et obtenons des fenêtres temporelles

$$X_k = (x_k, x_{k+1}, \dots, x_{k+s-1}) \quad k = 1, 2, \dots, N - s + 1.$$

Chaque segment X_k est transformé en un graphe de visibilité horizontale. Nous obtenons donc une série de graphes ou, de façon équivalente, de matrices d'adjacences $G = \{g_1, g_2, \dots, g_{k+s-1}\}$. À partir de ces dernières, nous construisons un nouveau réseau R dont les nœuds sont les états distincts, i.e. les matrices distinctes de G . Deux nœuds i et j de R sont reliés si la matrice g_i est directement suivie dans le temps par la matrice g_j . Le réseau R ainsi obtenu est pondéré et dirigé.

Nous appliquons cela à Rio de Janeiro en considérant une taille de fenêtre $s = 5$. Chaque segment est donc composé de 5 données, ce qui représente environ une durée d'un mois. En outre, si nous prenons $s = 4$, nous n'avons que 4 possibilités de réseaux, ce qui est restrictif. À l'inverse, si nous prenons $s = 6$, nous avons trop de possibilités par rapport

au nombre de données que nous possédons. Le réseau résultant possède 8 états différents correspondant aux 8 graphes de visibilité différents. Ces derniers sont illustrés à la figure 6.2.

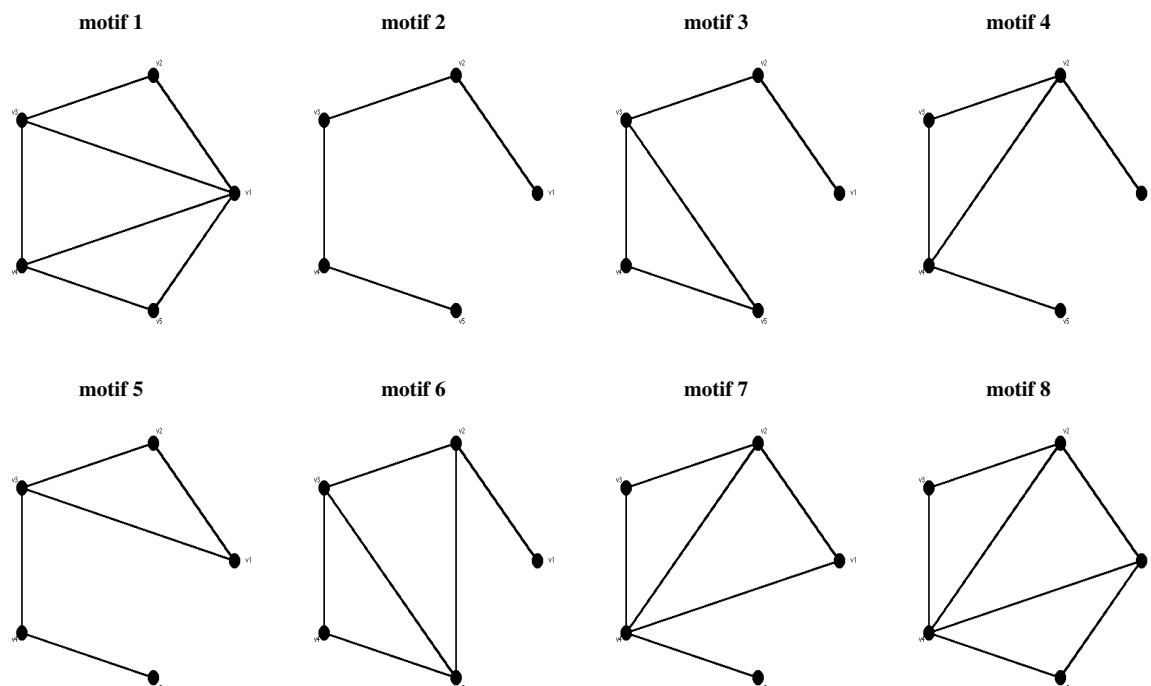


FIGURE 6.2 – Les différents motifs présents dans le réseau R de Rio pour $s = 5$.

Le réseau R est repris à la figure 6.3. Chaque nœud possède un label a/b qui signifie que le nœud est associé au motif a et ce motif est apparu pour la première fois dans la suite $G = \{g_1, g_2, \dots, g_{k+s-1}\}$ à la place b . Les nœuds et liens possèdent une taille proportionnelle à leur degré et leur poids respectivement. Le poids des liens montre le fait que certains états tendent à évoluer vers un autre état bien spécifique. Par exemple, le nœud 5/11 a tendance à évoluer le plus souvent vers le nœud 2/2 qui correspond au cas où chaque donnée ne voit que ses plus proches voisins. Nous pouvons également remarquer deux boucles de quatre motifs, à savoir $2/2 \rightarrow 6/15 \rightarrow 7/16 \rightarrow 5/11 \rightarrow 2/2$ et $2/2 \rightarrow 3/9 \rightarrow 4/10 \rightarrow 5/11 \rightarrow 2/2$.

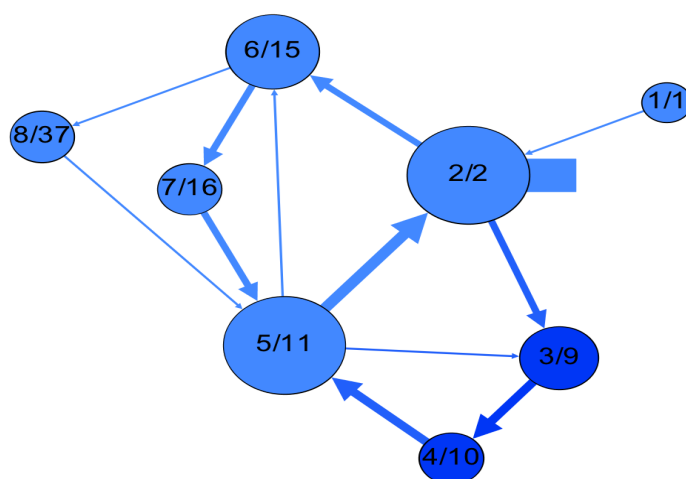


FIGURE 6.3 – Réseau R pour la municipalité de Rio de Janeiro en considérant $s = 5$.

En vue de "prédire" les motifs suivants, nous pouvons effectuer une marche aléatoire sur ce réseau. Évidemment, nous n'avons que des motifs basés sur le nombre de personnes infectées ; en pratique, cette marche aléatoire n'a pas beaucoup de sens (voire aucun) puisqu'il faudrait prendre en compte d'autres facteurs qui varient d'une année à l'autre tels que le taux d'humidité ainsi que les températures. Cependant, d'un point de vue "irréel", nous pouvons reformer une série de motifs. Pour pouvoir reformer une série temporelle, il faudrait être capable de travailler avec une suite de

graphes de visibilité ; chaque graphe correspond à plusieurs signaux et donc combiner une séries de graphes n'est pas si trivial. Cela pourrait être intéressant dans le cas de séries temporelles de grande taille puisque le nombre de motifs à 5 nœuds est restreint.

Nous avons également que le motif 2 est le motif le plus présent dans la série. Nous aimerions savoir si il apparaît dans la série de façon aléatoire ou non. Pour ce faire, nous utilisons la *rescaled range* qui est une mesure statistique permettant de détecter des persistances dans des séries temporelles [6]. Soit une série temporelle. Le *rescaled range* de cette dernière est le rapport entre R , la plus grande différence entre les données de la série et s , l'écart type des données de cette série. Si nous considérons cette même série temporelle mais avec plus d'observations, le *rescaled range* augmente aussi. Une façon de représenter cette augmentation est de tracer le graphe du logarithme de R/s en fonction du logarithme de n . En outre, la pente de cette droite est une estimation de l'exposant de Hurst.

En pratique, nous reformons une série avec les positions du motif le plus fréquent, à savoir le motif 2. À la figure 6.4 est représentée la relation R/s en fonction de n de la série reformée. La pente de la droite estimant au mieux les points est de 0.59, ce qui suggère que le motif possède un comportement persistant le long de la série temporelle de base. Il est évident que nous ne possédons pas beaucoup de données et que ce résultat doit être pris très "à la légère" mais d'une certaine façon, cela n'est pas étonnant puisque le virus est dépendant de certains paramètres qui ne sont clairement pas aléatoires.

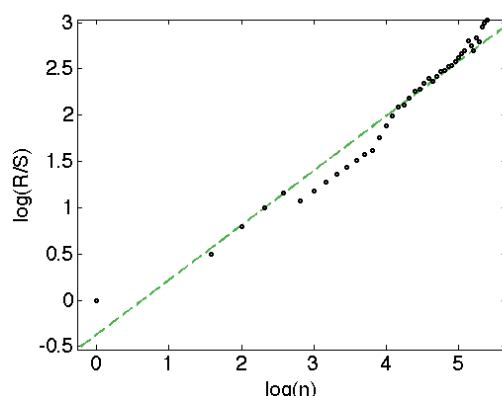
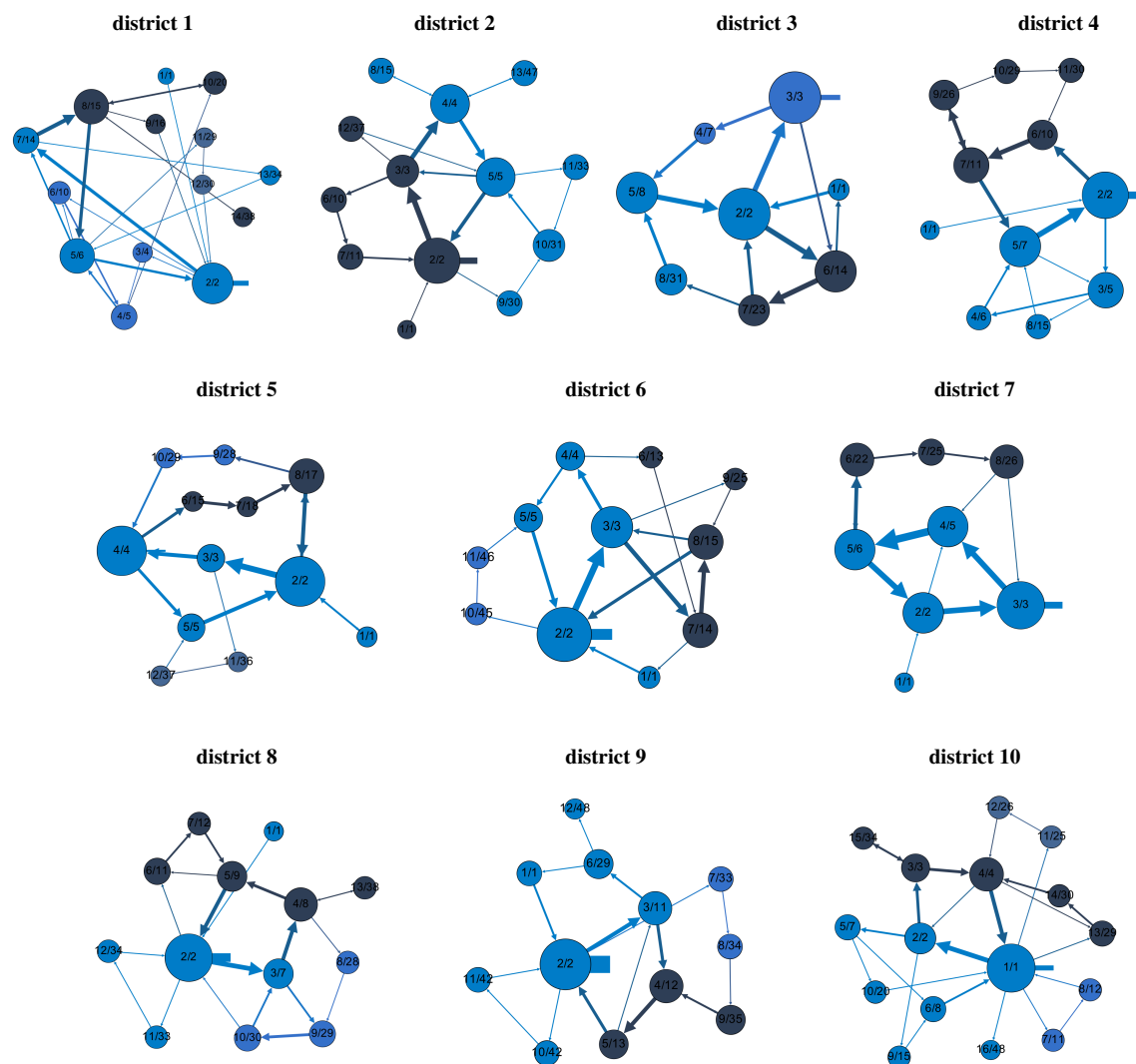


FIGURE 6.4 – Relation R/s en fonction de n pour la série formée à partir des positions du motif le plus présent dans la série temporelle de base. En noir sont les données, la ligne verte représente la droite estimant au mieux les données.

Districts

Nous avons appliqué la même "procédure" que pour la municipalité de Rio (figure 6.5) sur les 10 districts.

Le nombre de motifs uniques varient entre 8 (pour les districts 3 et 7) et 16 (pour le district 10). À première vue, quel que soit le district considéré, le motif apparaissant en deuxième position est un état fort présent dans la série.

FIGURE 6.5 – Les réseaux R obtenus en considérant $s = 5$ des 10 districts.

Lorsque nous considérons tous les districts, nous remarquons qu'il n'y a, en réalité, que 20 motifs différents. Ces derniers sont repris à la figure 6.6 (*remarque* : les numéros ne correspondent pas aux labels des nœuds des graphes de la figure 6.5). La figure 6.7 indique l'évolution de chaque district à travers les 48 fenêtres temporelles considérées. Pour chaque fenêtre temporelle, chaque district est identifié par un carré de couleur correspondant à un motif. Nous voyons clairement que le motif numéro 7 est dominant. Il correspond au cas où chaque donnée ne voit que ses plus proches voisins. En particulier, la figure 6.8 reprend le nombre d'occurrences de chaque motif. À l'instar de Rio entier, le motif 10 suit souvent le motif 7 (lorsque ce dernier ne se suit pas lui-même). En effet, dans 54 % des cas, le motif 7 est suivi par lui-même, dans 33% par le motif 10, dans 9 % par le motif 13, dans 2% par le 11, dans 1% fois par le 15 et enfin, dans le dernier 1% par le 19.

Lorsque nous mélangeons les données (à l'intérieur d'un même district), les résultats changent. Il n'y a plus de motif super présent comme avant (le motif 7 est deux fois plus présent que le second plus présent : 144 à 67). Cette fois, il y a, en moyenne, 4 motifs plus présents que les autres. Leur occurrence varie de 65 à 75, ce qui est beaucoup moins que le 144 du motif 7. Notons tout de même que le motif correspondant à "les données ne voient que leurs plus proches voisins" est présent dans les 4 motifs-hubs.

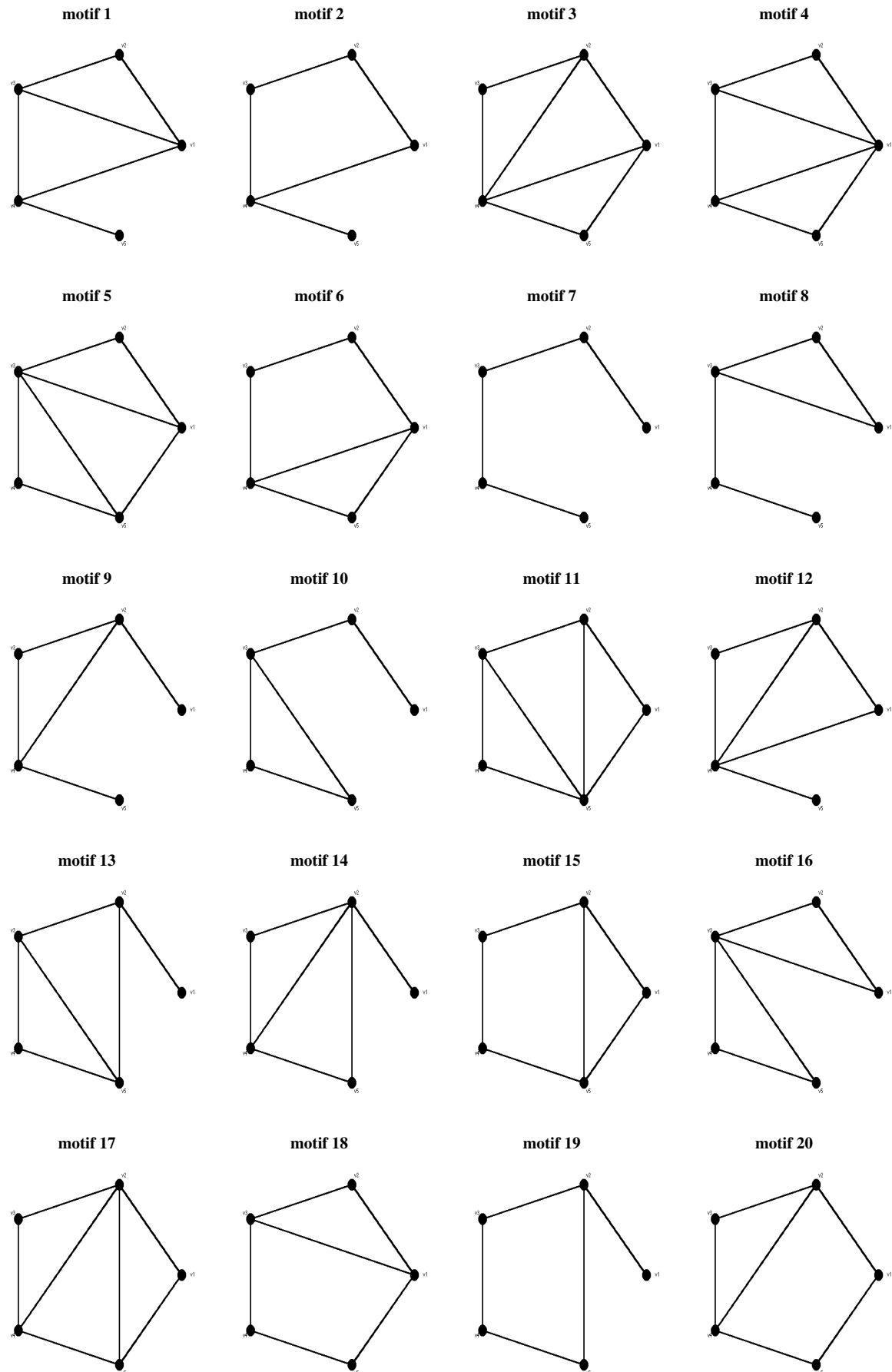


FIGURE 6.6 – Les 20 graphes de visibilité différents sur tous les districts en considérant $s = 5$.

Les districts ayant le plus de "ressemblance" à partir de la figure 6.7 sont les districts 1, 2, 3, 4 et 7. Il s'agit des districts en bord de mer à l'exception du district 3. Néanmoins, ce district n'est pas enfoncé à l'intérieur des terres comme

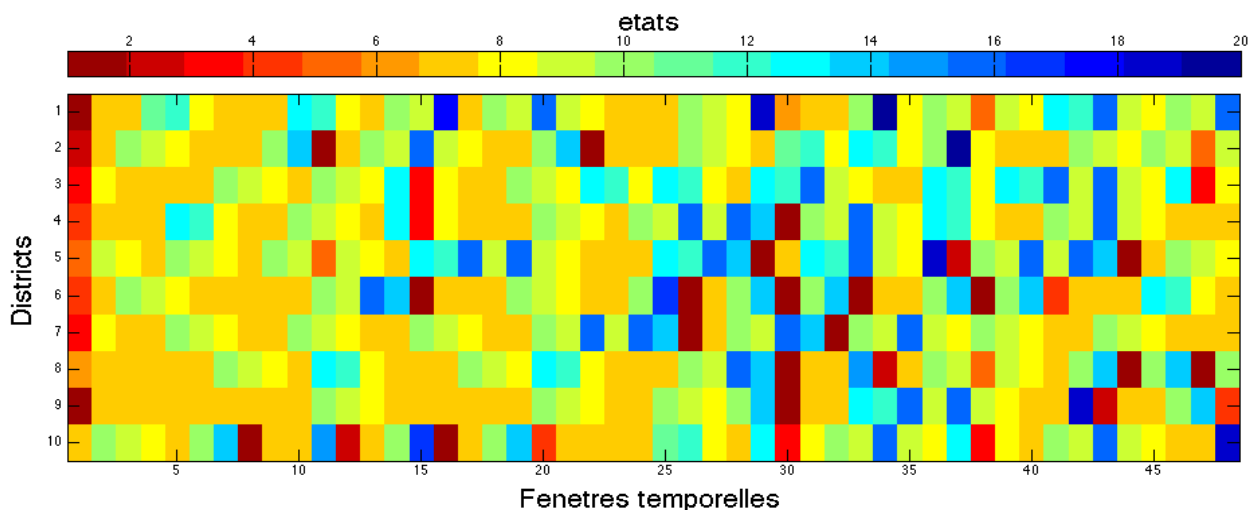


FIGURE 6.7 – Evolution de chaque district sur les 48 fenêtres temporelles. Chaque couleur correspond à un motif (figure 6.6).

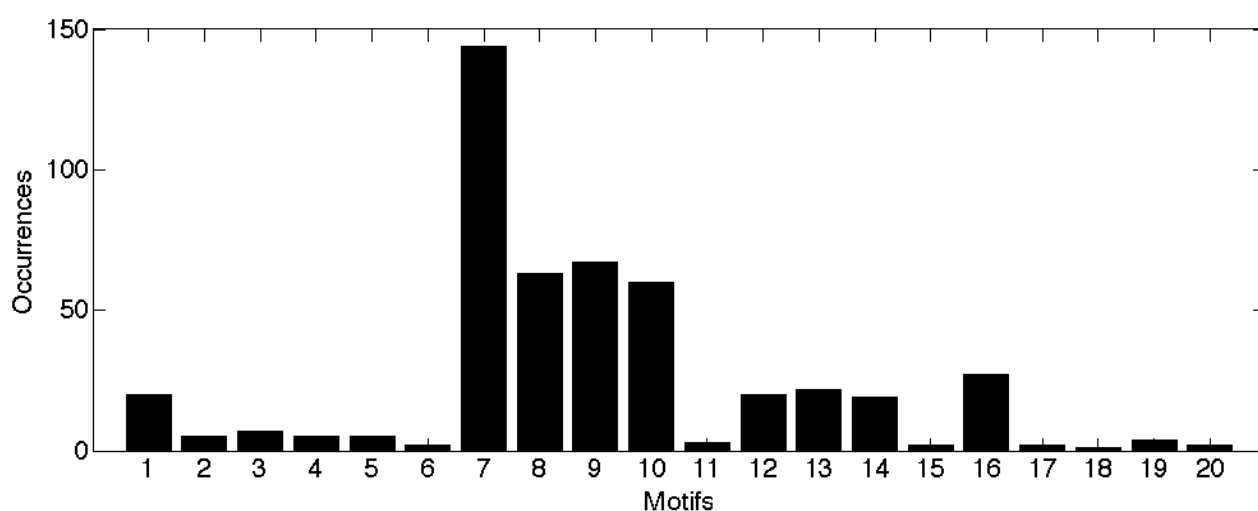


FIGURE 6.8 – Occurrences de chaque motif lors de l'évolution des districts (figure 6.7).

les sont les 5 autres restants. Lorsque nous parlons de ressemblance, nous ne regardons pas uniquement les motifs mais plutôt la signification des motifs : un motif représente plusieurs sous-séries qui peuvent être fort différentes. Dès lors, à la place de prendre directement les motifs, nous faisons correspondre chaque sous-série de 5 données à un vecteur ordonné de 5 symboles : 1, 2, 3, 4 et 5 (la plus grande donnée est assignée à 1, la seconde plus grande à 2, ...). Au final, nous obtenons une matrice de dimension $5 \times 48 \times 10$ et comparons chaque slice de cette matrice (vecteur unique, fréquence, ...).

Nous pouvons à présent regarder les corrélations entre ces districts. La figure 6.9 reprend, pour chacune des 10 fenêtres temporelles, les corrélations calculées au moyen de l'information mutuelle entre les 5 districts. Notons que nous ne pouvons utiliser le taux d'information mutuelle car les séries temporelles sont trop courtes : le temps de décorrélation n'est pas "calculable".

Nous pouvons considérer ces matrices de corrélations comme étant des matrices d'adjacence de graphes pondérés. Par simplicité, nous fixons un seuil α et lorsque la corrélation entre deux districts est inférieure à ce seuil, nous supprimons le lien. Sinon, nous gardons le lien avec un poids de 1. Nous obtenons donc finalement dix matrices d'adjacence binaires. En vue de regarder des similitudes temporelles, nous faisons varier le seuil α dans l'intervalle $[0.1 ; 0.5]$. Nous remarquons que les corrélations entre la première semaine temporelle est la même que pour la neuvième. Il en va de même pour les fenêtres temporelles 6 à 8.

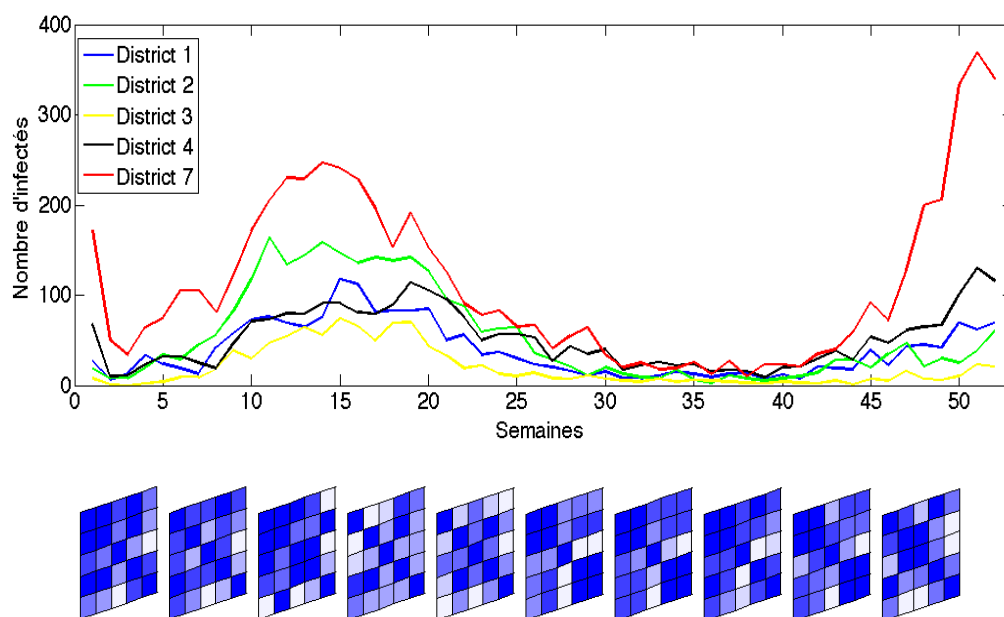


FIGURE 6.9 – Évolution des corrélations entre les 5 districts situés le plus en bordure de mer. En haut : les 5 séries temporelles correspondant au nombre de personnes infectées par le virus de la dengue par district. En bas : pour chaque fenêtre temporelle correspondante de longueur $s = 5$, la matrice de corrélation des districts calculée par l'information mutuelle.

Enfin, nous avons également testé la méthode DCI sur ces données. Comme le montre la figure 6.10, les résultats diffèrent de ceux obtenus précédemment. Nous avons généré les systèmes homogènes de deux façons : (a) en mélangeant les données par district ainsi que (b) sur tous les districts. Les résultats sont un peu différents bien que le meilleur CRS soit le même dans les deux cas. Les districts 3 et 8 ne sont jamais repris des ces sous-ensembles. Les valeurs obtenues pour la méthode (a) sont également moins élevées qu'en (b). Notons tout de même qu'il n'y a pas de différence très nette entre les valeurs des Tcs (pour une même méthode) ; le faible nombre de données ne donne sûrement pas des résultats très fiables.

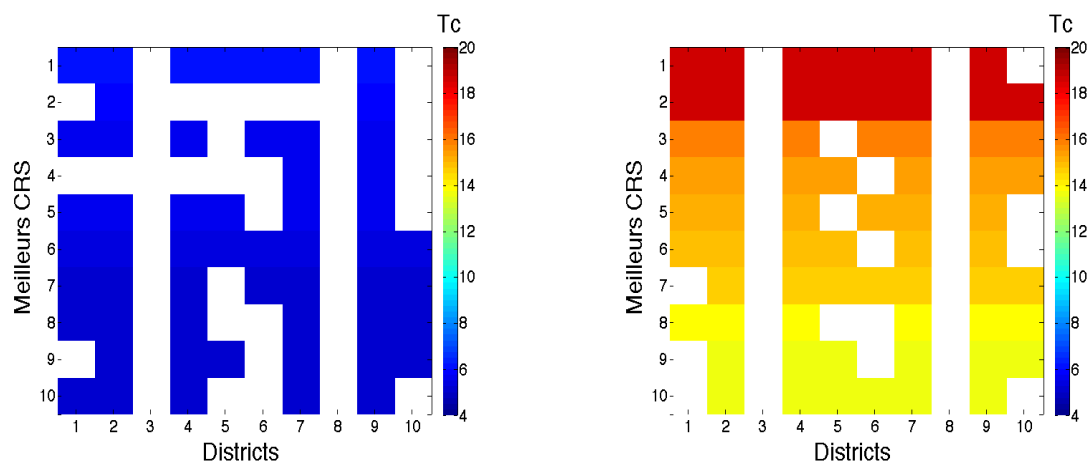


FIGURE 6.10 – Les 10 sous-ensembles possédant le meilleur Tc calculés par la méthode DCI. Les scores sont obtenus en moyennant sur 10 réalisations de systèmes homogènes en mélangeant les données (a) par districts ; (b) sur tous les districts.

6.2 Données cerveau

Dans cette seconde section, les données traitées correspondent aux signaux de 90 zones cérébrales pendant 140 unités de temps. Nous nous intéressons en particulier aux corrélations existantes entre ces 90 zones.

Corrélation entre les séries temporelles

Dans un premier temps, nous calculons le taux d'information mutuelle entre chaque paire de zones cérébrales. Comme nous l'avons vu dans un chapitre précédent, ce calcul requiert un partitionnement de l'espace, i.e. définir un nombre de bins n pour le calcul des probabilités. La figure 6.11 reprend les matrices de corrélations pour différentes valeurs de n ainsi que la matrice de corrélation obtenue par Pearson. Bien que les résultats soient évidemment très sensibles au nombre de bins, les panels (a) et (b) montrent 2 ensembles de zones plus corrélées entre elles qu'avec le reste : les zones 22 à 28 ainsi que 61 à 67. Le panel (c), calculé par Pearson, reconnaît les deux ensembles mais propose d'autres zones fortement corrélées. À noter que les échelles ne sont pas les mêmes puisque $\text{coeff}_{\text{Pearson}} \in [-1, 1]$ tandis que $\text{MI} \in [0, 1]$.

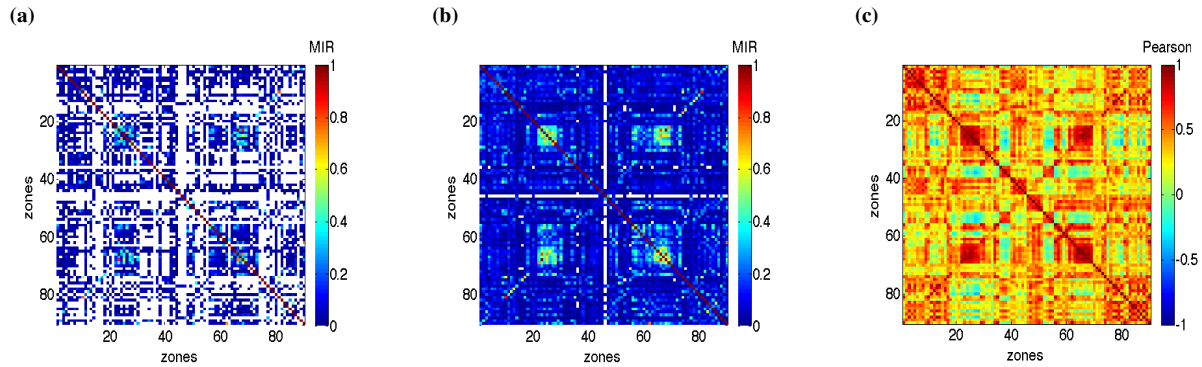


FIGURE 6.11 – Matrices de corrélation pour les 90 zones cérébrales (a)-(b) calculées avec le taux d'information mutuelle avec 4 et 17 bins respectivement ; (c) calculée par Pearson.

Puisque nous sommes contraints de choisir le nombre de bins et que ce choix ne semble pas si évident, une idée serait de calculer le taux d'information mutuelle entre les graphes de visibilité associées aux signaux. En particulier, chaque série temporelle est transformée en un graphe de visibilité horizontale. Nous obtenons donc 90 graphes et calculons les corrélations entre ces derniers par le taux d'information mutuelle entre chaque paire de distributions de degrés. Plus précisément, nous calculons

$$\text{MI}_{x,y} = \sum_{a_x} \sum_{b_y} p(a_x, b_y) \log_2 \frac{p(a_x, b_y)}{p(a_x)p(b_y)}$$

où x et y sont deux graphes de visibilité horizontale, $p(a_x, b_y)$ est la probabilité jointe qu'un nœud soit de degré a dans le graphe x et de degré b dans le graphe y . Afin d'obtenir le taux d'information mutuelle, nous appliquons la même méthode qu'au chapitre 4. Au plus $\text{MI}_{x,y}$ est élevée, au plus les distributions de degrés entre les graphes sont semblables et par conséquent, la structure des séries temporelles.

Cela est la même chose qu'auparavant, en ce sens que nous "symbolisons" la série temporelle, à l'exception que nous ne contrôlons plus le nombre de symboles. Ce dernier est déterminé par la visibilité qu'ont les données. La figure 6.12 illustre la matrice de corrélation obtenue sur les distributions de degrés par le taux d'information mutuelle ainsi que par Pearson. Quelle que soit la méthode de calcul, nous trouvons toujours les deux ensembles fortement corrélés (22-28 et 61-67) mais également d'autres ensembles tels que 2-18 et 74-90. Cela n'est pas sans rappeler les ensembles trouvés par Pearson sur les séries directement.

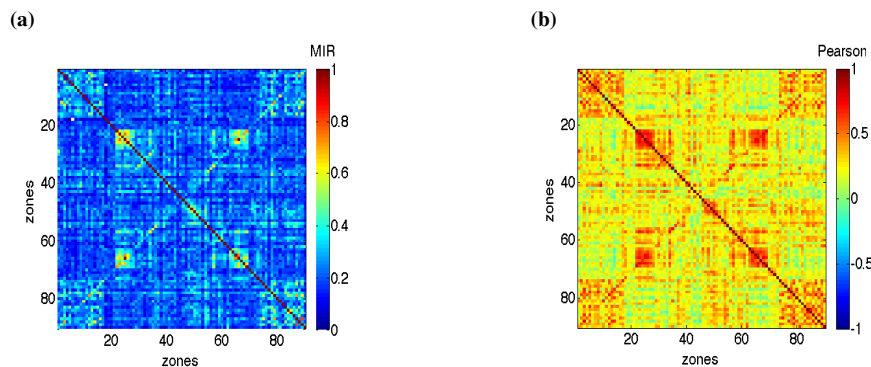


FIGURE 6.12 – Matrices de corrélation pour les 90 zones cérébrales à partir des distributions de degrés des graphes de visibilité horizontale (a) calculées avec le taux d'information mutuelle ; (d) calculée par Pearson.

L'idée de remplacer la série temporelle par la distribution de degrés du graphe correspondant est motivée par le fait

que la valeur (hauteur) d'une donnée est reliée au degré du nœud correspondant. Le degré maximal d'un nœud pour ces différents signaux est égal à 17. Nous remarquons pourtant que la matrice de la figure 6.11(b) est fort différente de celle présente à la figure 6.12(a). Nous avons également vu que le degré moyen d'un nœud (d'un HVg) était compris entre 2 et 4 et que la borne supérieure était atteinte pour des séries aléatoires. Au vu de l'allure des séries temporelles, il ne semble donc pas si saugrenu de vouloir comparer les figures 6.11(a) et 6.12(a) ; nous n'observons pourtant que quelques similitudes. En outre, le taux d'information mutuelle moyen est beaucoup plus élevé entre les graphes qu'entre les signaux correspondants (figure 6.13). Nous ne savons pas lequel est le meilleur, mais cela permet de calculer des corrélations entre séries temporelles par des mesures de graphes.

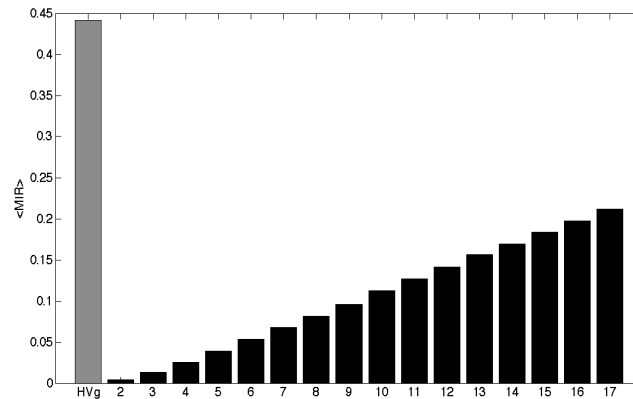


FIGURE 6.13 – Taux d'information mutuelle moyen entre les 90 zones cérébrales lorsqu'il est calculé à partir des graphes de visibilité (HVg) ainsi que des séries temporelles (les chiffres indiquent le nombre de bins pris pour le calcul). Le taux d'information mutuelle moyen semble être proportionnel au nombre n de bins.

Une autre façon de regarder les corrélations entre les différentes zones du cerveau tout en utilisant les graphes de visibilité est de s'intéresser non plus aux degrés des nœuds mais aux liens présents dans chacun des graphes de visibilité. Nous calculons donc les corrélations entre les matrices d'adjacence de ces graphes. Étant donné que ce sont des matrices binaires, nous n'avons pas besoin de "symboliser" les données et pouvons calculer directement le taux d'information mutuelle entre les différentes matrices d'adjacence. Pour ce faire, nous considérons les matrices de dimension $m \times n$ comme des vecteurs de dimension $1 \times mn$. Le résultat est représenté à la figure 6.14 ; ce sont à nouveau les zones 22-28 et 61-67 qui partagent le plus d'information entre elles et remarquons également, à l'instar de la figure 6.12, une corrélation un peu plus prononcée dans les zones 2-18 et 74-90.

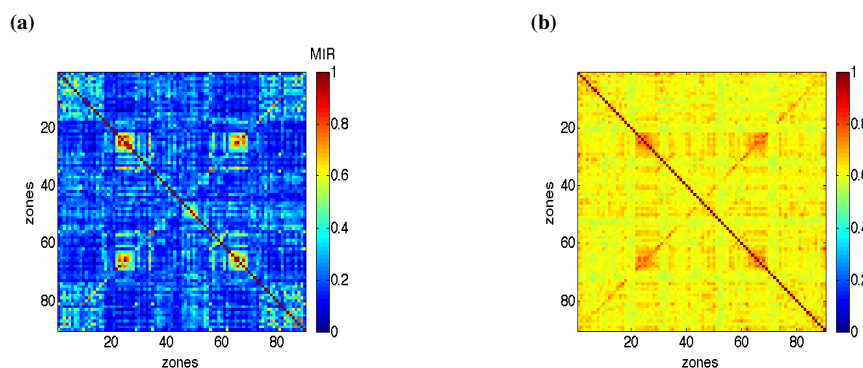


FIGURE 6.14 – Matrice (a) du taux d'information mutuelle ; (b) de corrélation de Pearson entre les 90 zones du cerveau. Ces mesures sont calculées à partir des 90 matrices d'adjacence des graphes de visibilité.

Au vu des matrices 6.12(a) et 6.14(a), une question naturelle nous vient à l'esprit : quelle information gagnons-nous en considérant la matrice d'adjacence à la place des degrés ? De façon générale, pour un réseau formé de N nœuds, une matrice d'adjacence contient $N \times N$ éléments alors que la suite des degrés n'en contient que N . Dès lors, la question devient : existe-il une bijection entre la matrice d'adjacence d'un HVg et la série de degrés correspondante ? Il a été démontré dans [33] que cela était le cas lorsque la série temporelle était canonique (i.e. une série $S = \{x_1, x_2, \dots, x_N\}$ à valeurs réelles telle que x_i et x_N soient les plus grandes données et qu'il n'y ait pas de doublons dans les autres données, i.e. $x_i \neq x_j \forall i, j = 2, \dots, N-1$). Nos données ici ne sont pas des séries canoniques mais il est possible de les transformer pour qu'elles le soient. En particulier, si x_1 et x_N ne sont pas les 2 plus grandes valeurs de la série, nous cherchons l'indice m tel que x_m soit la plus grande valeur de la série. Nous construisons ensuite une extension de la série $S^* = \{x_1, x_2, \dots, x_m, \dots, x_N, x_1, \dots, x_m, \dots, x_N\}$ et extrayons de cette dernière la sous-série de $N+1$ données $S^c = \{x_m, \dots, x_m\}$ qui est donc la série canonique associée à S . Malheureusement, dans notre cas, les indices m associés

aux différents signaux ne sont pas égaux et donc nous ne pouvons restreindre, par cet argument, les corrélations entre les matrices d'adjacence aux corrélations entre les degrés.

Évolution temporelle

Nous regardons à présent l'évolution temporelle des différents signaux. En particulier, nous considérons des fenêtres de quatorze unités de temps disjointes. Pour chacune d'elle, nous calculons le graphe de visibilité horizontale associé à chaque signal et calculons l'information mutuelle entre chaque couple de distributions de degrés. Nous pouvons considérer la matrice d'information mutuelle comme la matrice de poids d'un graphe comportant 90 nœuds représentant les 90 zones cérébrales. Nous calculons enfin, pour chacune des matrices obtenues, l'information mutuelle moyenne. La figure 6.15 reprend cette valeur pour chaque fenêtre temporelle. Pour les valeurs les plus "extrêmes", nous regardons ce qu'il se passe au niveau des réseaux. Les réseaux représentés sont les réseaux obtenus par la matrice d'information mutuelle avec seulement quelques liens. En réalité, nous avons pris les 90 nœuds et les liens utiles les plus lourds pour que les graphes soient connexes. La taille des nœuds est proportionnelle au degré de ces derniers. Nous remarquons, sur notre exemple, que lorsque l'information mutuelle moyenne est "élevée", les réseaux possèdent un nœud "central", i.e. avec un plus grand degré. À l'inverse, lorsque l'information mutuelle moyenne est plus faible, il n'y a pas de nœud dominant.

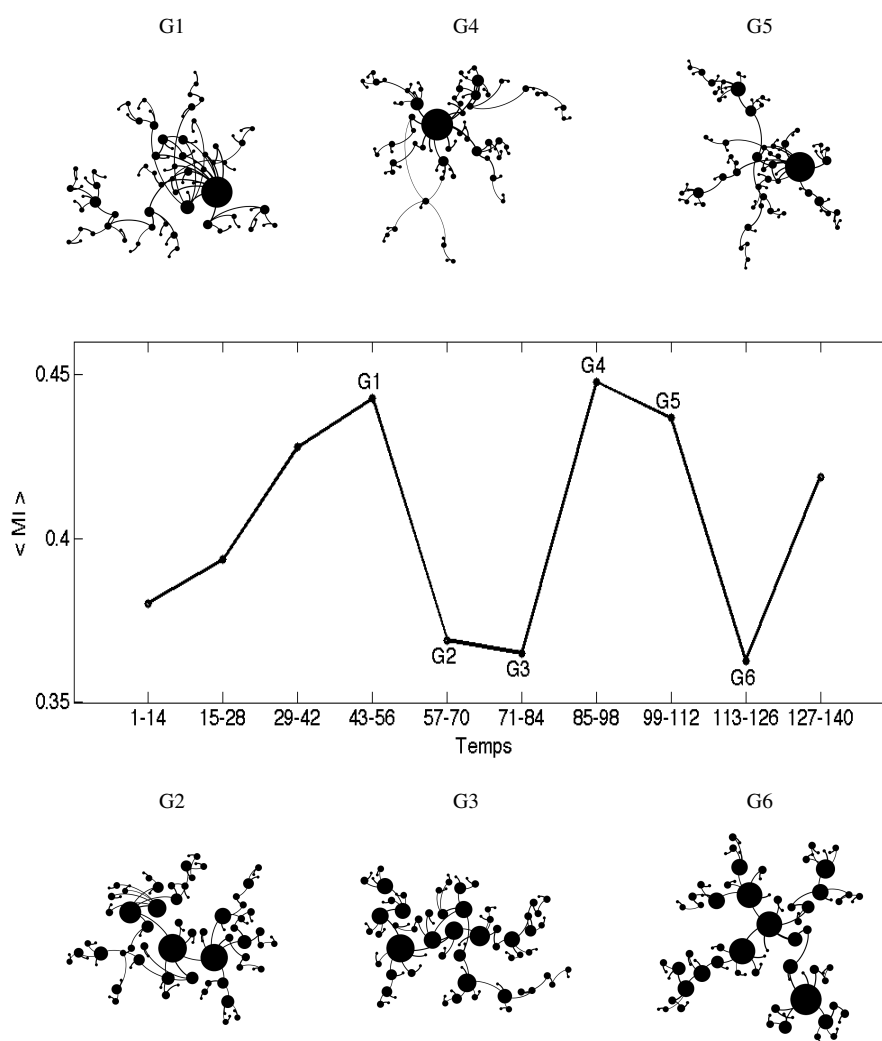


FIGURE 6.15 – Évolution de l'information mutuelle moyenne pour chaque fenêtre temporelle. Cette mesure est calculée à partir des distributions de degrés des graphes de visibilité associés à chaque signal. Les graphes associés aux valeurs de $mean(MI)$ les plus importantes possèdent un nœud central, ce qui n'est pas le cas lorsque $mean(MI)$ est faible.

Bien que la longueur des fenêtres temporelles soit très courte (14), il semblerait que cela permette de capturer certaines informations. Il serait possible que les valeurs de l'information mutuelle moyenne associées à différents moments aient un rapport avec la structure sous-jacente des différents réseaux. Dès lors, il serait intéressant de comprendre ce qu'il se passe pendant ces laps de temps particuliers, peut-être directement à partir des séries temporelles.

Nous regardons à présent l'évolution des degrés de chaque nœud, comme illustré à la figure 6.16. Cela nous permet d'observer qu'il n'y a que 3 fenêtres temporelles où un nœud possède un degré beaucoup plus important que les autres. Ces 3 fenêtres temporelles correspondent aux moments où l'information mutuelle moyenne est au plus haut (figure

6.15). Nous remarquons également que ces hubs de la figure 6.15 ne sont pas les mêmes nœuds/zones. Cependant, ils appartiennent tous les trois aux ensembles identifiés précédemment, à savoir 61-67 et 74-90. Pour les autres fenêtres temporelles, les degrés de nœuds semblent être distribués de façon plus homogène.

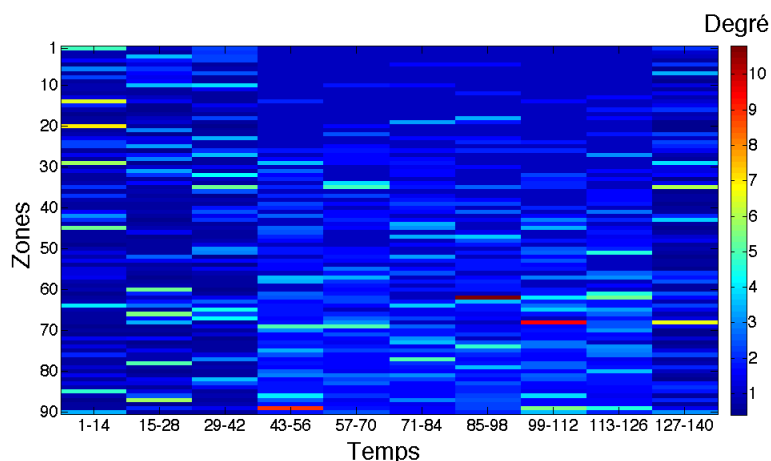


FIGURE 6.16 – Évolution des degrés des nœuds des graphes de visibilité horizontale. Les fenêtres 43-56, 85-98 et 99-112 possèdent un hub (rectangle rouge foncé), contrairement aux autres où les degrés semblent distribués de façon plus homogène.

Finalement, nous nous intéressons aux communautés présentes dans ces réseaux. Ces derniers sont tous modulaires : la modularité de ces derniers oscille entre 0.7595 et 0.8026 ; et possèdent entre 8 et 11 modules. Pour une représentation plus visuelle, la table 6.1 est représentée à la figure 6.17. Ce type de diagramme permet de représenter les changements dans la structure du réseau dans le temps (dans notre cas). Chaque "colonne" représente un graphe de visibilité correspondant à une fenêtre temporelle possédant plusieurs modules caractérisés par des rectangles noirs. Ces modules sont classés par tailles (haut-bas) et au plus le module contient de nœuds, au plus la hauteur du rectangle noir correspondant est grande. Enfin, les liens colorés entre les colonnes représentent les "flux", i.e. les changements dans la composition des graphes dans le temps. Bien que cette représentation ne permette pas de déterminer quels nœuds appartiennent à quels modules, nous remarquons que ces derniers ne sont pas formés des mêmes nœuds (ce qui nous savions déjà puisque le nombre de modules varie d'une fenêtre temporelle à l'autre). Cela permet surtout de voir comment les nœuds d'une communauté vont évoluer (au sens des communautés). En outre, si un nœud appartient à la communauté la plus importante au "temps" t , il n'appartiendra pas forcément à la communauté la plus importante au "temps" $t + \Delta t$. Par exemple, pour la fenêtre temporelle 57-70, aucun des nœuds présents dans la communauté la plus importante n'était déjà présent dans le module le plus important (au "temps" 43-56) et aucun d'entre eux en le sera au "temps" 71-84.

Temps	Modularité	Nb. de comm.	Tailles des communautés
1-14	0.7595	9	[22, 12, 10, 9, 9, 9, 8, 7, 4]
15-28	0.7600	9	[17, 15, 2, 10, 9, 8, 7, 6, 6]
29-42	0.7799	8	[15, 14, 14, 13, 11, 10, 9, 4]
43-56	0.7667	10	[14, 12, 12, 12, 12, 8, 6, 6, 4, 4]
57-70	0.8026	10	[12, 10, 10, 10, 10, 10, 8, 8, 8, 4]
71-84	0.7956	11	[12, 10, 10, 10, 10, 8, 8, 6, 6, 6, 4]
85-98	0.7849	11	[14, 10, 10, 8, 8, 8, 8, 6, 6, 6, 6]
99-112	0.7885	11	[16, 12, 10, 10, 8, 6, 6, 6, 6, 6, 4]
113-126	0.8023	10	[12, 12, 10, 10, 8, 8, 8, 8, 8, 6]
127-140	0.7677	8	[17, 15, 14, 11, 11, 10, 8, 4]

TABLE 6.1 – Résumé de l'"analyse" de la modularité des graphes de visibilité.

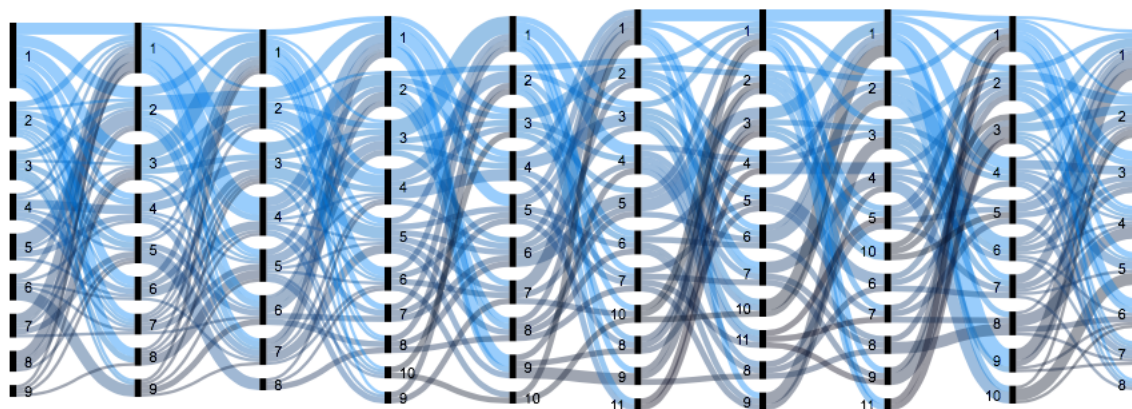


FIGURE 6.17 – Représentation des flux entre les différentes communautés présentes dans les graphes de visibilité.

Conclusion

Dans ce chapitre, nous avons tenté d'appliquer les méthodes expliquées dans les chapitres précédents sur des données "réelles". Nous avons pu constater que, malgré le nombre restreint de données, certaines de ces méthodes, en particulier les graphes de visibilité, étaient applicables. En ce sens, nous avons évoqué de travailler avec des sous-graphes de visibilité pour caractériser une série temporelle. Ce n'est clairement pas au point, mais cela pourrait être approfondi et amélioré. Ce chapitre a également permis de montrer que certaines méthodes implémentées dans ce travail ne sont pas encore prêtes à être utilisées "telles qu'elles". En l'occurrence, la méthode DCI qui, comme écrite ici, ne permet de traiter que de très petits systèmes ; elle est totalement incapable de fournir, pour les données relatives au cerveau, des résultats pertinents en un temps acceptable.

Conclusion

Ce travail se basait principalement sur quatre articles qui ont pour objectif de caractériser et de comprendre le fonctionnement d'un système. Dans les chapitre 2 et 3, nous avons développé deux méthodes permettant de transformer une série temporelle en réseau. Nous avons montré pour la méthode basée sur les quantiles qu'il était possible de définir une transformation inverse : reconstruire une série temporelle à partir d'un réseau mais que le transfert des propriétés d'un objet à l'autre nous était encore obscur. Quant à la seconde méthode traitant des graphes de visibilité, nous avons esquissé quelques moyens de faire le lien entre les séries et graphes associés. Ce lien se fait via la distribution de degrés des graphes. Les chapitres 5 et 6 se focalisaient plus sur l'organisation d'un système. En particulier, le chapitre 5 proposait une méthode pour détecter les liens entre les composants d'un système via le taux d'information mutuelle. Quant au chapitre 6, il propose d'utiliser le *dynamical cluster index* pour identifier des sous-ensembles candidats "importants" dans l'organisation d'un système. Quelle que soit la méthode (MIR ou DCI), le point le plus épineux concerne le calcul des probabilités lorsque les signaux considérés sont à valeurs réelles. Le dernier chapitre a d'ailleurs permis de pointer ce problème et a suggéré le couplage de certaines méthodes pour éviter le choix du nombre de symboles nécessaires pour le calcul des probabilités. De façon plus générale, ce dernier chapitre a permis de se rendre compte que les méthodes développées dans le cadre de ce travail nécessitent plusieurs ajustements et approfondissements.

Au niveau des ajustements, nous pensons principalement à la méthode DCI et à son temps d'exécution beaucoup trop important que pour traiter des systèmes "réels" (i.e. autres que des systèmes "modèles" comme proposés dans la chapitre 5). Afin de pouvoir considérer de tels systèmes, il faudrait être capables de gérer la pléthore de sous-ensembles possibles en en sélectionnant que quelques uns de façon judicieuse. Une façon telle que proposée dans [55] est d'utiliser des algorithmes génétiques. Quant aux approfondissements, commençons par la méthode des quantiles. Comme déjà mentionné, la façon dont les propriétés sont transférées des séries aux réseaux est encore inconnue ; il serait intéressant de mettre une "théorie" derrière tout ça. Nous avons également proposé d'utiliser des chaînes de Markov d'ordre supérieur mais le choix de la représentation des graphes dans ce cadre (i.e. une quantile - un nœud ou un couple de quantiles - un nœud) a certainement de l'importance quand aux propriétés présentes dans les signaux et conservées dans les réseaux. Concernant les graphes de visibilité, nous avons évoqué au dernier chapitre de travailler avec des motifs afin d'analyser le comportement temporel d'une série. Il serait intéressant de s'attarder dessus, ce qui permettrait peut-être de caractériser des séries par ces sous-graphes. Nous n'avons aussi considéré, dans le dernier chapitre, que les graphes de visibilité horizontale. Ce choix a été motivé par le fait qu'il s'agit d'une simplification de la visibilité naturelle. Il serait dès lors intéressant de comparer les résultats obtenus avec les graphes de visibilité naturelle. Cette liste de propositions n'est évidemment pas exhaustive ; il existe bien sûr beaucoup d'autres directions qui valent la peine d'être considérées.

Références

- [1] P. ABRY et F. SELLAN : *The wavelet-based synthesis for the fractional Brownian motion proposed by F. Sellan and Y. Meyer : Remarks and fast implementation*. Appl. and Comp. Harmonic Anal., 1996.
- [2] P.S. ADDISON : *Fractals and Chaos : An Illustrated Course*. Institute of Physics Publishing, 1997.
- [3] R. BAGGIO et R. SAINAGHI : *Mapping time series into networks as a tool to assess the complex dynamics of tourism systems* . Tourism Management, 54:23 – 33, 2016.
- [4] J.-M. BARDET, I. KAMMOUN et V. BILLAT : *A new process for modeling heartbeat signals during exhaustive run with an adaptive estimator of its fractal parameters*. Journal of Applied Statistics, 39(6):1331–135, 2012. <http://dx.doi.org/10.1080/02664763.2011.646962>.
- [5] J.-M. BARDET, G. LANG, G. OPPENHEIM, A. Philippe and S. STOEV et M.S. TAQQU : *Generators of long-range dependence processes : a survey*.
- [6] James B. BASSINGTHWAIGHT et Gary M. RAYMOND : Evaluating rescaled range analysis for time series. *Annals of Biomedical Engineering*, 22(4):432–444, 7 1994.
- [7] M. BASTIAN, S. HEYMANN et M. JACOMY : *Gephi : An Open Source Software for Exploring and Manipulating Networks*. <http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154>, 2009.
- [8] E. BIANCO-MARTINEZ, N. RUBIDO, C.G. ANTONOPOULOS et M.S. BAPTISTA : *Successful network inference from time-series data using Mutual Information Rate*. Chaos, 2016.
- [9] V.D. BLONDEL, J.-L. GUILLAUME, R. LAMBIOTTE et E. LEFEBVRE : *Fast unfolding of communities in large networks*. Journal of Statistical Mechanics : Theory and Experiment, vol. 2008, no 10, 2008.
- [10] A. BUTTE et I. S. KOHANE : *Mutual information relevance network : Functional genomic clustering using pairwise entropy measurements*. Pac. Symp. Biocomput., 5, 2000.
- [11] A.S.L.O. CAMPANHARO, M.I. SIRER, R.D. MALMGREN, F.M. RAMOS et L.A.N. AMARAL : *Duality between Time Series and Networks*. PloS ONE 6(8) : e23378, 2011.
- [12] A. CARBONE : *Algorithm to estimate the Hurst exponent of high-dimensional fractals*. Phys. Rev. E, 76:056703, Nov 2007.
- [13] G. A. DARBELLAY et I. VAJDA : *Estimation of the information by an adaptive partitioning of the observation space*. IEEE Transactions on Information Theory, pages 1315–1321, 1999.
- [14] T. DIEKER : *Simulation of fractional Brownian motion*.
- [15] D. MINGZHOU et Y. WEIMING : *Distribution of the first return time in fractional Brownian motion and its application to the study of on-off intermittency*. Phys. Rev. E, 52:207–213, Jul 1995.
- [16] Ministério do meio AMBIENTE : <https://grupounidas.wordpress.com/2009/06/27/mapa-da-producao-do-lixo-no-municipio-do-rio-de-janeiro/>, consulté le 27 mars 2017.
- [17] J.P. ECKMANN, S.O. KAMPHORST et D. RUELLE : *Recurrence Plot of Dynamical Systems*. Europhysics Letters 5 (9) : 973–977, 1987.
- [18] A.M. FRASER et H.L. SWINNEY : *Independent coordinates for strange attractors from mutual information*. Phys. Rev. A, 33:1134–1140, Feb 1986.
- [19] Thomas M. J. FRUCHTERMAN et Edward M. REINGOLD : *Graph Drawing by Force-directed Placement*. *Softw. Pract. Exper.*, 21(11):1129–1164, novembre 1991.
- [20] M. FU : *Hurst parameter estimation on fractional Brownian motion and its application to the development of the zebrafish*, 2013.
- [21] M. P. GOLOMBEK : *Assessment of Mars Exploration Rover landing site predictions*. Nature, 436:44–48, 2005.
- [22] H. HERZEL, A.O. SCHMITT et W. EBELING : *Finite sample effects in sequence analysis*. Chaos, Solitons & Fractals, 4(1):97 – 113, 1994.
- [23] J.M. JOYCE : *Kullback-Leibler Divergence*, pages 720–722. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [24] Jan W. KANTELHARDT : *Fractal and Multifractal Time Series*, pages 463–487. Springer New York, New York, NY, 2011.

- [25] A. KRASKOV, H. STÖGBAUER et P. GRASSBERGER : *Estimating mutual information*. Phys. Rev. E, 69:066138, Jun 2004.
- [26] S. KUZNETSOV et A. KUZNETSOV : *Classic Arnold's cat and other maps on a torus*. <http://www.sgtnd.narod.ru/eng/main.htm>, consulté le 26 janvier 2017.
- [27] L. LACASA, B. LUQUE, F. BALLESTROS, J. LUQUE et J.C. NUNO : *From time series to complex networks : The visibility graph*. Proc Natl Acad Sci USA 105, 2008.
- [28] L. LACASA, B. LUQUE, J. LUQUE et J.C. NUNO : *The visibility graph : A new method for estimating the Hurst exponent of fractional Brownian motion*. EPL (Europhysics Letters), 86(3):30001+, 2009.
- [29] L. LACASA, A. NUÑEZ, E. ROLDÁN, J. M. R. Parrondo B. et LUQUE : *Time series irreversibility : a visibility graph approach*. The European Physical Journal B, 85(6):217, 2012.
- [30] L. LACASA et R. TORAL : *Description of stochastic and chaotic series using visibility graphs*, 2010.
- [31] R. LAMBIOTTE : *Théorie des graphes*, 2015-2016. Université de Namur.
- [32] E. N. LORENZ : *Deterministic Nonperiodic Flow*. Journal of Atmospheric Sciences, 20:130–148, mars 1963.
- [33] Bartolo LUQUE et Lucas LACASA : Canonical horizontal visibility graphs are uniquely determined by their degree sequence. *The European Physical Journal Special Topics*, 226(3):383–389, 2017.
- [34] H.A. MAKSE, S. HAVLIN, M. SCHWARTZ et H.E. STANLEY : *Method for generating long-range correlations for large systems*. pre, 53:5445–5449, mai 1996.
- [35] B.B. MANDELBROT et J. W. Van NESS : *Fractional Brownian Motions, Fractional Noises and Applications*. SIAM Rev, 1968.
- [36] N. MARWAN, J.F. DONGES, Y. ZOU, R.V. DONNER et J. KURTHS : *Complex Network Approach for Recurrence Analysis of Time Series*. Physics Letters A 46, 2009.
- [37] Y. MOON, B. RAJAGOPALAN et U. LALL : *Estimation of mutual information using kernel density estimators*. Phys. Rev. E, 52, 1995.
- [38] E. NELSON : *Dynamical Theories of Brownian Motion*. Princeton University Press, 2001.
- [39] M. E. J. NEWMAN : *Power laws, Pareto distributions and Zipf's law*. *Contemporary Physics*, 46:323–351, septembre 2005.
- [40] M.E.J. NEWMAN : *Modularity and community structure in networks*. Proc. Natl. Acad. Sci. USA, vol. 103, no 23, p. 8577-8582, 2006.
- [41] A.M. NUÑEZ, B. LUQUE, J.P. GOMEZ et L. LACASA : *Visibility Algorithms : A Short Review*. INTECH Open Access Publisher, 2012.
- [42] A. NUÑEZ, L. LACASA, E. VALERO, J. GÓMEZ et B. LUQUE. : *Detecting Series Periodicity with Horizontal Visibility Graphs*. I. J. Bifurcation and Chaos, 22(7), 2012.
- [43] S. OKAMURA : *The Short Time Fourier Transform and Local Signals*. Thèse de doctorat, Carnegue Mellon University, Pittsburgh, Pennsylvania 15213, 2011.
- [44] E. PADOUE : *Chaîne de Markov et génome*. <https://www.i2m.univ-amu.fr/pardoux/enseignement/MG.pdf>, consulté le 14 décembre 2016.
- [45] S. PARTHASARATHY, S. MEHTA et S. SRINIVASAN : *Robust Periodicity Detection Algorithms*. Proceedings of the 15th ACM international conference on Information and knowledge management.
- [46] C.-K. PENG, S. V. BULDYREV, A. L. GOLDBERGER, S. HALVIN, F. SCIORTINO, M. SIMONS et H. E. STANLEY : *Long-range correlations in nucleotide sequences*. Nature, 356:168–170, 1992.
- [47] N. RUBIDO, E. BIANCO-MARTINEZ, C. GREBOGI, M.S. BAPTISTA et C. MASOLLER : *Exact detection of direct links in networks of interacting dynamical units*. New J. Phys. 16, 093010, 2014.
- [48] G. B. RYBICKI : *Notes on Gaussian Random Functions with Exponential Correlation Functions (Ornstein-Uhlenbeck Process)*.
- [49] O.E. RÖSSLER : *An equation for continuous chaos*. Physics Letters A, 57(5):397 – 398, 1976.
- [50] D.W. SCOTT : *On optimal and data-based histograms*. Biometrika, 66, 605-610, 1979.
- [51] S.H. STROGATZ : *Nonlinear Dynamics And Chaos : With Applications To Physics, Biology, Chemistry, And Engineering*. Westview Press ; 1 édition, 2001.
- [52] H. STURGE : *The choice of class-interval*. J. Amer. Statist. Assoc., 21, 65–66, 1926.
- [53] T.W.KÖRNER : *Fourier Analysis*. illustrée, réimprimée édition, 1998.
- [54] S. VAJAPPEYAM : *Understanding Shannon's Entropy metric for Information*. CoRR, 2014.
- [55] M. VILANI, A. ROLI, A. FILISSETTI, Irene POLI et R. SERRA : *The search for candidate relevant subsets of variables in complex systems*. Artificial Life, 2015.
- [56] C. WANG et X. Sean WANG : *Supporting Content-based Searches on Time Series via Approximation*. International Conference on Scientific and Statistical Database Management, 2000.

- [57] D. J. WATTS et S. H. STROGATZ : *Collective dynamics of 'small-world' networks*. Nature, 1998.
- [58] W.W.S. WEI : *Time Series Analysis : Univariate and Multivariate Methods*. Pearson Addison Wesley, 2006.
- [59] G. WEISS : *Time-reversibility of linear stochastic process*. J. Appl. Prob., 12:831–836, 1975.
- [60] WIKIPEDIA : *Morlet wavelet*. https://en.wikipedia.org/wiki/Morlet_wavelet. consulté le 9 mai 2016.
- [61] WIKIPEDIA : *Freedman–Diaconis rule*. https://en.wikipedia.org/wiki/Freedman–Diaconis_rule, 2016. consulté le 4 novembre 2016.
- [62] WIKIPEDIA : *Histogram*. <https://en.wikipedia.org/wiki/Histogram>, 2016. consulté le 4 novembre 2016.
- [63] WIKIPEDIA : *Mouvement brownien*, 2016.
- [64] WIKIPEDIA : *Kullback–Leibler divergence*. https://en.wikipedia.org/wiki/Kullback–Leibler_divergence, 2017. consulté le 25 janvier 2017.
- [65] H. WILLIAM et al. : *Power Spectrum Estimation Using the FFT*. sec. 13.4, Numerical Recipes in C, 2nd ed., Cambridge University Press, 1992.
- [66] A. WOLF, J.B. SWIFT, H.L. SWINNEY et J.A. VASTANO : *Determining Lyapunov Exponents from a Time Series*. Physica 16D (1985)285-317, 1984.
- [67] X. XU, J. ZHANG et M. SMALL : *Superfamily phenomena and motifs of networks induced from time series*. PNAS, 105(50), 2008.
- [68] Z.-M. YIN : *New Methods for Simulation of Fractional Brownian Motion*. Journal of computational physics, 127, 1996.
- [69] J. ZHANG et M. SMALL : *Complex Network from Pseudoperiodic Time Series : Topology versus Dynamics*. Physical Review Letters, 96, 238701, 2006.
- [70] D. ZHOU, Y. ZHANG, Y. XIAO et D. CAI : *Reliability of the Granger causality inference*. New Journal of Physics, 16(4):043016, 2014.

Annexes

Nous présentons dans ces annexes les principales fonctions élaborées dans le cadre de ce travail.

Fonctions relatives au chapitre 2

```
1 function [W,M2,qx] = S2N(T,Q,x,q,Norder)
2
3 % BUT : Calculer la matrice d'adjacence du graphe associé à une série
4 % temporelle
5 % Inputs : T, le nombre de données dans x
6 %          Q, le nombre de quantiles
7 %          x, la série temporelle
8 %          q, les quantiles
9 %          Norder, l'ordre pour la "mémoire" de la chaîne de Markov
10 % Outputs : W, la matrice d'adjacence pour Norder == 1
11 %           M2, la matrice d'adjacence pour Norder > 1. Si Norder == 1, M2
12 %             = 0
13 %           qx, x symbolisé (par les quantiles)
14 % Source : A.S.L.O. CAMPANHARO, M.I. SIRER, R.D. MALMGREN, F.M. RAMOS et
15 %          L.A.N. AMARAL : Duality between Time Series and Networks. PloS ONE 6(8)
16 %          : e23378, 2011.
17
18 for i = 1:T
19     for l = 1:Q
20         if (l == 1)
21             if (q(l) <= x(i)) && (x(i) <= q(l+1))
22                 qx(i) = l;
23                 break
24             end
25         elseif (q(l) < x(i)) && (x(i) <= q(l+1))
26             qx(i) = l;
27             break
28         end
29     end
30 end
31
32 W = getTransitionMatrix(qx,1);
33
34 M2 = 0;
35
36 if Norder>1
37     M2 = getTransitionMatrix(qx,Norder);
38 end
39 end
40
41 function b = N2S(T,Q,M,q,Norder)
42
43 % BUT : Calculer la série temporelle associée au réseau
44 % Inputs : T, le nombre de données dans x
45 %          Q, le nombre de quantiles
46 %          M, le réseau
```

```

7 %           q, les quantiles
8 %           Norder, l'ordre pour la "mémoire" de la chaîne de Markov
9 % Outputs : b, la série temporelle associé au réseau M
10 % Source : A.S.L.O. CAMPANHARO, M.I. SIRER, R.D. MALMGREN, F.M. RAMOS et
    L.A.N. AMARAL : Duality between Time Series and Networks. PloS ONE 6(8)
    : e23378, 2011.
11 % Remarque : Implémentation très "grossière" : automatiser pour toutes les
    valeurs des Norders
12
13 j = 1;
14
15 switch Norder
16     case 1
17         h = randi([1 Q],1,1);
18         while (j <= T)
19             test = M(h,1);
20             jet = unifrnd(0,1);
21             k = 1;
22             while(jet>test && k<Q)
23                 k = k+1;
24                 test = test + M(h,k);
25             end
26             h = k;
27             b(j) = q(h)+abs(q(h+1)-q(h))*rand(1);
28             j = j+1;
29         end
30
31     case 2
32         p = randi([1 Q],1,1);
33         c = randi([1 Q],1,1);
34         while (j <= T)
35             i = (p-1)*Q+c;
36             %M(i,:) = M(i,:)/sum(M(i,:));
37             test = M(i,1);
38             jet = unifrnd(0,1);
39             s = 1;
40             while(jet>test && s<Q)
41                 s = s+1;
42                 test = test + M(i,s);
43             end
44             b(j) = q(s)+abs(q(s+1)-q(s))*rand(1);
45             p = c;
46             c = s;
47             j = j+1;
48         end
49
50     case 3
51         pp = randi([1 Q],1,1);
52         p = randi([1 Q],1,1);
53         c = randi([1 Q],1,1);
54         while (j <= T)
55             i = (pp-1)*Q^2+(p-1)*Q+c;
56             %M(i,:) = M(i,:)/sum(M(i,:));
57             test = M(i,1);
58             jet = unifrnd(0,1);
59             s = 1;
60             while(jet>test && s<Q)
61                 s = s+1;
62                 test = test + M(i,s);
63             end
64             b(j) = q(s)+abs(q(s+1)-q(s))*rand(1);
65             pp = p;
66             p = c;

```

```

67         c = s;
68         j = j+1;
69     end
70 end

1  function [Ar,indices,cost_init,cost] = reorder(A,iter)
2
3  % But : Réordonner une matrice de telle sorte que la fonction de poids
4  %       (éq. (2.1) dans le texte) soit minimisée
5  % Inputs : A, la matrice à réordonner
6  %          iter, le nombre de d'itérations
7  % Outputs : Ar, la matrice A réordonnée
8  %           indices, les indices pour retrouver les permutations
9  %           cost_init, le coût (fonction de poids) de A
10 %           cost, le coût (fonction de poids) de Ar
11
12 N = length(A);
13
14 dist = toeplitz(0:N-1,0:N-1);
15 best = sum(sum(dist.*A))/N;
16 cost_init = best;
17
18 Astart = A;
19 start_i = 1:N;
20
21 for j=1:iter
22     i = 1:N;
23     r = randperm(N);
24     i(r(1)) = r(2);
25     i(r(2)) = r(1);
26     new_cost = sum(sum(A(i,i).*dist))/N;
27     if (new_cost < best)
28         A = A(i,i);
29         tmp = start_i(r(2));
30         start_i(r(2)) = start_i(r(1));
31         start_i(r(1)) = tmp;
32         best = new_cost;
33     end
34 end
35 Ar = Astart(start_i,start_i);
36 indices = start_i;
37 cost = best;

```

Fonction relative au chapitre 3

```

1  function [W,Wout] = Visibility(x, m)
2
3  % But : Obtenir la/les matrice(s) d'adjacence (in et out) du graphe de
4  %       visibilité naturelle/horizontale (dirigé)
5  % Input : x, la serie temporelle à transformer
6  %         m, la méthode : 'natural','horizontal','directedhorizontal'
7  % Outputs : W, la matrice d'adjacence du graphe de visibilité
8  %           naturelle/naturelle (in du horizontale dirigé) associe à x
9  %           Wout, (facultatif - en fonction de la méthode) la matrice
10 %          d'adjacence out du graphe de visibilité naturelle associe à x
11 % Source : A.M. NUÑEZ, B. LUQUE, J.P. GOMEZ et L. LACASA : Visibility
12 %          Algorithms : A Short Review. INTECH Open Access Publisher, 2012.
13
14 l = length(x);
15
16 W = zeros(l);

```



```

14 if nargout == 1
15     if strcmp(m,'natural')
16         for i=1:l-1
17             W(i,i+1) = 1;
18             W(i+1,i) = 1;
19             for j=i+2:l
20                 cpt = 0;
21                 for n = i+1:j-1
22                     if x(n) < (x(i)+(x(j)-x(i))*(n-i)/(j-i))
23                         cpt = cpt+1;
24                     else
25                         break;
26                     end
27                 end
28                 if cpt == j-i-1
29                     W(i,j) = 1;
30                     W(j,i) = 1;
31                 end
32             end
33         end
34     elseif strcmp(m,'horizontal')
35         for i=1:l-1
36             W(i,i+1) = 1;
37             W(i+1,i) = 1;
38             for j=i+2:l
39                 cpt = 0;
40                 for n = i+1:j-1
41                     if (x(i)>x(n)) && (x(j)>x(n))
42                         cpt = cpt+1;
43                     else
44                         break;
45                     end
46                 end
47                 if cpt == j-i-1
48                     W(i,j) = 1;
49                     W(j,i) = 1;
50                 end
51                 if x(j) >= x(i)
52                     break
53                 end
54             end
55         end
56     end
57 elseif nargout == 2 && strcmp(m,'directedhorizontal')
58     Wout = zeros(l);
59     for i=1:l-1
60         Wout(i,i+1) = 1;
61         W(i+1,i) = 1;
62         for j=i+2:l
63             cpt = 0;
64             for n = i+1:j-1
65                 if (x(i)>x(n)) && (x(j)>x(n))
66                     cpt = cpt+1;
67                 else
68                     break;
69                 end
70             end
71             if cpt == j-i-1
72                 Wout(i,j) = 1;
73                 W(j,i) = 1;
74             end
75             if x(j) >= x(i)
76                 break

```

```

77         end
78     end
79 end
80 else
81     error('En argument: la série temporelle + ''natural'', ''horizontal'',
82         ou ''directedhorizontal''. En sortie : Un ou deux "objets"')
83 end
84 end

```

Fonctions relatives au chapitre 4

```

1 function [mirb,mib] = MutInfRate(x)
2
3 % But : Calculer le taux d'information mutuelle entre chaque paire de
4 % composants d'un système
5 % Input : x, le système contenant les séries temporelles
6 % Outputs : mirb, vecteur contenant mutual information rate entre chaque
7 %           paire de composants du système x
8 %           mib, vecteur contenant mutual information entre chaque paire de
9 %           composants du système x
10 % Source : E. BIANCO-MARTINEZ, N. RUBIDO, C.G. ANTONOPOULOS et M.S.
11 %          BAPTISTA : Successful network inference from time-series data using
12 %          Mutual Information Rate. Chaos, 2016.
13
14 M = size(x,1);
15 N = [10];
16 su = 0;
17 suu = 0;
18 mir = zeros(1,M*(M-1)/2);
19 z = zeros(1,M*(M-1)/2);
20
21 for ec = 1:length(N)
22     o = 1;
23     for k = 1:M-1
24         for l = k+1:M
25             xy = [x(k,:);x(l,:)];
26             xy = RD(xy,N(ec));
27             hxy = entrop(xy);
28             hx = entrop(xy(1,:));
29             hy = entrop(xy(2,:));
30             z(o) = hx+hy-hxy;
31
32             [a,~,c] = unique(xy','rows');
33             G = zeros(length(a(:,1)));
34             for i = 1:length(c)-1
35                 G(c(i),c(i+1)) = 1;
36             end
37
38             dis = allpaths(G);
39             d = max(dis(:));
40
41             mir(o) = z(o)/d;
42
43             o = o+1;
44         end
45     end
46
47     mirh = (mir-min(mir))/(max(mir)-min(mir));
48     mih = (z-min(z))/(max(z)-min(z));
49     su = su+mirh;
50     suu = suu+mih;
51 end

```

```

47 end
48
49 mirb = su/max(su);
50 mirb = mirb';
51 mib = suu/max(suu);
52 mib = mib';
53 end
54
55
56 % Fonctions secondaires :
57 function new = RD(x,nbins)
58
59 % But : Symboliser une série temporelle
60 % Inputs : x, la série temporelle à symboliser
61 %          nbins, le nombre de symboles
62 % Output : new, x symbolisé
63
64 for i = 1:size(x,1)
65
66     xmin = round(min(x(i,:)));
67     xmax = round(max(x(i,:)));
68     seg = (xmax-xmin)/nbins;
69
70     if seg == 0
71         new = zeros(size(x));
72     else
73         partition = (xmin+seg):seg:(xmax-seg);
74         new(i,:) = quantiz(x(i,:),partition,(0:1:nbins-1))+1;
75         new(i,:) = quantiz(x(i,:),partition)+1;
76     end
77 end
78 end
79
80 function H = entrop(x)
81
82 % But : Calculer l'entropie d'un système
83 % Input : x, le système dont nous voulons l'entropie
84 % Output : H, l'entropie de x
85
86 x = sortrows(x')';
87 n = length(x(1,:));
88 i = 1;
89 H = 0;
90 while i<=n
91     j = 1;
92     while (i+j<=n) && all(x(:,i)==x(:,i+j))
93         j = j+1;
94     end
95     if i==n
96         H = H-(1/n)*log2(1/n);
97     else
98         H = H-(j/n)*log2(j/n);
99     end
100    i = i+j;
101 end
102 end

```

Fonctions relatives au chapitre 5

```

1 function tab = MainDCI(A)
2
3 % But : Calculer les indices Tc, DCI pour les sous-ensembles d'un système

```

```

4  % Input : A, système dont nous voulons les sous-ensembles "pertinents"
    pour sa compréhension
5  % Output : tab, matrice contenant par ligne le Tc, DCI pour un
    sous-ensemble (défini par les colonnes restantes), triée par ordre
    décroissant
6  % Source : M. VILANI, A. ROLI, A. FILISETTI, Irene POLI et R. SERRA : The
    search for candidate relevant subsets of variables in complex systems.
    Artificial Life, 2015.
7  % Remarque : À choisir : nbins, le nombre de symboles si séries à valeurs
    réelles
8  %                                     f, fraction de possibilités à évaluer
9
10 nbins = ...;
11 N = size(A,1);
12 x = RD(A,nbins);
13 H1 = zeros(1,N);
14 for i=1:N
15     H1(i) = entrop(x(i,:));
16 end
17 HS = entrop(x);
18
19 xh = homogeneous(x,'full');
20 H1h = zeros(1,N);
21 for i=1:N
22     H1h(i) = entrop(xh(i,:));
23 end
24 HSh = entrop(xh);
25
26 f = ...;
27
28 tab = clust(x,xh,H1,H1h,HS,HSh,2,round(f*nchoosek(N,2)));
29 tabtmp = sortrows(tab,2);
30 if N > 2
31     for i=3:N-1
32         tabtmp =
33             clust(x,xh,H1,H1h,HS,HSh,i,round(f*nchoosek(N,i)));%,tabtmp(end,:));
34         tab = [tab ; tabtmp];
35         tabtmp = sortrows(tabtmp,2);
36     end
37 end
38 tab = sortrows(unique(tab,'rows'),-1);
39 end
40
41 % fonctions secondaires : (+ RD(x,nbins) et entrop(x), voir précédemment)
42 function Uh = homogeneous(U,m)
43
44 % But : Créer un système homogène à partir d'un système de base
45 % Input : U, le système de base
46 %         m, la méthode : 'full' ou 'part'
47 % Output : Uh, le système homogène associé à U
48
49 sU = size(U);
50
51 if strcmp(m,'full')
52     nU = numel(U);
53     T = reshape(1:nU,sU);
54     T(randperm(nU)) = T;
55     Uh = U(T);
56 elseif strcmp(m,'part')
57     for i = 1:sU(1)
58         Uh(i,:) = U(i,randperm(sU(2)));
59     end

```

```

60 else
61     error('En argument: la matrice + ''full'' ou ''part''')
62 end
63 end
64
65 function tab = clust(no,nh,H1,H1h,HS,HSh,t,ns,best)
66
67 % But : Calculer les indices Tc, DCI pour les sous-ensembles d'un système
68 % Input : no, le système de base
69 %         nh, le système homogène associé à no
70 %         H1, les entropies des éléments de no
71 %         H1h, les entropies des éléments de nh
72 %         HS, l'entropie du système no
73 %         HSh, l'entropie du système nh
74 %         t, la taille des sous-ensembles à tester
75 %         ns, le nombre de sous-ensembles à tester
76 %         best, (facultatif) le meilleur sous-ensemble (au sens du DCI) de
        taille t-1 si nous voulons une heuristique supplémentaire
77 % Output : tab, matrice contenant par ligne le Tc, DCI pour un
        sous-ensemble (défini par les colonnes restantes)
78 % Source : M. VILANI, A. ROLI, A. FILISETTI, Irene POLI et R. SERRA : The
        search for candidate relevant subsets of variables in complex systems.
        Artificial Life, 2015.
79
80 s = size(no);
81
82 indices = [1:s(1)];
83 nl = nchoosek(indices,t);
84 pp = 1;
85 for i=1:length(nl(:,1))
86     x = nh(nl(i,:),:);
87     HH1 = entrop(x);
88     Int(pp) = sum(H1h([nl(i,:)]))-HH1;
89     y = nh;
90     y(nl(i,:),:) = [];
91     HH2 = entrop(y);
92     Mut(pp) = HH1+HH2-HSh;
93     D(pp) = Int(pp)/Mut(pp);
94     pp = pp+1;
95 end
96 mD = mean(D);
97 sdD = std(D);
98 mInt = mean(Int);
99 mMut = mean(Mut);
100
101
102 pp = 1;
103 nl = nl(randperm(length(nl(:,1))),:);
104 for i=1:ns
105     clearvars x y;
106     x = no(nl(i,:),:);
107     HH1 = entrop(x);
108     Integration = sum(H1([nl(i,:)]))-HH1;
109     y = no;
110     y(nl(i,:),:) = [];
111     HH2 = entrop(y);
112     Mutual = HH1+HH2-HS;
113     DCI = (Integration)/(Mutual);
114     %DCIn = (Integration/mInt)/(Mutual/mMut);
115     TC = (DCI-mD)/sdD;
116     a = zeros(1,s(1)); a(nl(i,:))=1;
117     tab(pp,:) = [TC,DCI,[a]];
118     pp = pp+1;

```

```
119 end
120
121 if nargin == 9
122     for i=1:s(1)
123         vec = zeros(1,s(1));
124         vec(i) = 1;
125         tmp = mod(best(:,3:end)+vec,2);
126         if numel(nonzeros(tmp))>numel(nonzeros(best(:,3:end)))
127             f = find(tmp==1);
128             x = no(f',:);
129             HH1 = entrop(x);
130             Integration = sum(H1([f]))-HH1;
131             y = no(find(tmp==0)',:);
132             HH2 = entrop(y);
133             Mutual = HH1+HH2-HS;
134             DCI = (Integration)/(Mutual);
135             %DCIn = (Integration/mInt)/(Mutual/mMut);
136             TC = (DCI-mD)/sdD;
137             tab(pp,:) = [TC,DCI,[tmp]];
138             pp = pp+1;
139         end
140     end
141 end
142 end
```